U.PORTO

MESTRADO

SAÚDE PÚBLICA

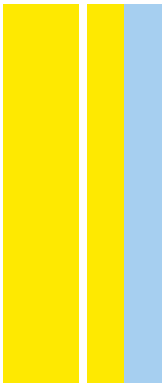
Monitoring morbidity associated with chronic conditions:

Study of the coexistence of chronic diseases and their impact on specific morbidity indicators in the Portuguese National Health Survey 2014

Ivo Cruz

M

2017



U.PORTO

Mestrado

Saúde Pública

Monitoring morbidity associated with chronic conditions:

Study of the coexistence of chronic diseases and their impact on specific morbidity indicators in the Portuguese National Health Survey 2014

Autor: Ivo Cruz

Orientador(a): Prof.ª Doutora Raquel Lucas
Faculdade de Medicina da Universidade do Porto
Instituto de Saúde Pública da Universidade do Porto

Porto

2017

Agradecimentos

O autor gostaria de agradecer a colaboração e o apoio dos restantes coautores pertencentes à equipa de investigação e desenvolvimento do presente trabalho, nomeadamente à Dr.ª Daniela Simões, Dr.ª Teresa Monjardino, Dr.ª Susana Barbosa, Dr. Luís Alves de Sousa, Dr. David Nascimento Moreira, Dr.ª Delfina Antunes e Prof.ª Doutora Raquel Lucas.

Gostaria de agradecer igualmente a colaboração do Instituto Nacional de Saúde Dr. Ricardo Jorge e do Instituto Nacional de Estatística, IP, pela cedência dos dados do 5º Inquérito Nacional de Saúde 2014.

Epígrafe

A presente dissertação de Mestrado em Saúde Pública, com o tema "*Monitoring morbidity associated with chronic conditions: Study of the coexistence of chronic diseases and their impact on specific morbidity indicators in the Portuguese National Health Survey 2014*", encontra-se organizada segundo as normas de publicação da revista científica *Journal of Clinical Epidemiology*, para a qual será proposta a sua publicação.

O trabalho entretanto desenvolvido foi apresentado em comunicação oral no *V Congresso Nacional de Saúde Pública*, a 16 de fevereiro de 2017, no Porto, Portugal, no *1º Encontro CESP para Apresentação e Discussão dos Protocolos de Investigação Epidemiológica – Do Projeto à Prática*, a 21 de abril de 2017, no Instituto de Saúde Pública da Universidade do Porto, no Porto, Portugal, e no Seminário *ROSNorte (Rede de Observatórios locais de Saúde do Norte): Observar a Saúde Localmente*, a 30 de junho de 2017, na Administração Regional de Saúde do Norte, IP, no Porto, Portugal. Foi igualmente aceite para comunicação oral na *10th European Public Health Conference*, a 02 de novembro de 2017, em Estocolmo, Suécia, estando agendada para o painel *3.K. – Controlling chronic conditions*.

Índice

Lista de figuras

Lista de tabelas

Lista de abreviaturas

DALY	Disability Adjusted Life Years
EHIS	European Health Interview Survey
NHS	National Health Survey
NUTS II	Level 2 of the Classification of Territorial Units for Statistics
PAF	Population Attributable Fraction
PC	Principal Component
PCA	Principal Component Analysis
PR	Prevalence Ratio
PR_{aj}	Adjusted Prevalence Ratio

Resumo

Objetivo

O estudo do impacto das doenças crónicas sobre a morbilidade é essencial para o planeamento em saúde e gestão de serviços de saúde. A coexistência natural de múltiplas doenças crónicas na população deve ser considerada na análise do seu impacto. Este estudo tem como objetivo analisar o impacto das doenças crónicas em indicadores específicos de morbilidade na população portuguesa, tendo em conta a sua coexistência.

Desenho de estudo

Foram usados os dados do Inquérito Nacional de Saúde português 2014. A coexistência de doenças crónicas foi estudada por uma análise de componentes principais, agrupando as doenças crónicas em componentes principais com uma correlação positiva e plausibilidade biológica para este agrupamento. As doenças crónicas iniciais foram reclassificadas em possíveis classes de causas suficientes para cada componente principal. As variáveis recodificadas foram modeladas utilizando um modelo de regressão log-Poisson multivariado. As exponenciais dos coeficientes de regressão foram utilizadas como uma medida da associação (razão de prevalência) para calcular as frações atribuíveis populacionais.

Resultados

Globalmente, a componente principal com maior impacto nos indicadores de morbilidade estudados foi a *artrose, lombalgia/cervicalgia, alergia e/ou depressão*. Considerando as doenças e coexistência das mesmas, foi igualmente a *artrose e lombalgia/cervicalgia*, isoladas ou coexistentes com outras doenças, as com maior impacto na morbilidade.

Conclusões

Para além da importância que o estudo do impacto das doenças crónicas na morbilidade tem para o planeamento em saúde e gestão de serviços de saúde baseados em evidência, a programação deste tipo de análise possibilita o seu uso na monitorização da saúde das populações.

Palavras-chave

Doença crónica; comorbilidade; inquéritos de saúde; análise de componentes principais; fração atribuível populacional.

Abstract

Objective

The study of the impact of chronic diseases on morbidity is essential for health planning and healthcare services management. Given the natural coexistence of multiple chronic diseases in the population, this must be taken into account in the analysis of their impact. This study aims to analyse the impact of chronic diseases on specific morbidity indicators in the Portuguese population, considering their coexistence.

Study design and setting

Data from the Portuguese National Health Survey 2014 were used. The coexistence of chronic disease was studied by a principal component analysis, grouping chronic diseases in principal components with a positive correlation and biological plausibility for this grouping. The initial chronic diseases were reclassified by possible classes of sufficient causes for each principal component. These recoded variables were modelled using a multivariate log-Poisson regression. The exponential of the coefficients of the regression were used as a measure of association (prevalence ratio) for calculating the population attributable fractions.

Results

Overall, the principal component with the greatest impact on the studied morbidity indicators was *arthrosis, low back/neck disorder, allergy and/or depression*. Considering the diseases and their coexistence, it was also *arthrosis and low back/neck disorder*, isolated or coexisting with other diseases that had the greatest impact on morbidity.

Conclusions

In addition to the importance of studying the impact of chronic diseases on morbidity to evidence based health planning and health services management, the programming of this type of analysis enables its use in monitoring the health of populations.

Keywords

Chronic disease; comorbidity; health surveys; principal component analysis; population attributable fraction.

1. Introduction

The study of chronic diseases and their impact on the health and well-being of populations is essential for health planning and for the organization and management of healthcare services. However, most of the available national and regional health information focuses mainly on mortality data. This information does not emphasize the importance to the health of populations of relevant chronic diseases with little impact on mortality, such as musculoskeletal, psychiatric or sensory organs diseases (1-3). On the other hand, the existing information on morbidity only uses frequency measures, such as incidence or prevalence data of diseases. Few studies evaluate the impact of diseases on morbidity with global indicators, such as disability adjusted life years (DALY) (2,3), or on specific indicators of morbidity. Also, the study of morbidity with DALY, a measure that attempts to summarize the health of a population by aggregating data on morbidity and mortality, does not allow evaluating the impact of chronic diseases on specific morbidity dimensions.

Recently, some studies have adopted methodologies similar to those proposed by Perruccio, *et al*. (4), in which the impact of chronic diseases on specific indicators of morbidity is estimated through population attributable fractions (PAF) (5-8). This analysis allows the estimation of the proportion of the morbidity indicator that could be prevented if the disease of interest was eliminated from the population, considering that the distribution of other diseases would remain unchanged and the theoretical assumptions of the PAF are verified. Another possible interpretation would be the proportional weight of the impact that a given disease has on the indicator in question. This analysis of diseases and their impact on population morbidity provides an essential complement to the information for evidence based health planning and for the organization and management of healthcare services.

However, in the calculation of PAF the natural coexistence of certain diseases in the population is not usually considered. The impact of each disease is estimated only adjusting to the existence of other diseases. If, on the one hand, this method allows studying each disease individually, taking into account the possible existence of others, it does not reflect the realistic coexistence of diseases in the population, nor does it analyse the impact of their coexistence on morbidity. As multimorbidity is a growing public health problem and has a great impact on population health and health systems, it is essential to better identify the patterns of multimorbidity when studying chronic diseases and their impact on health outcomes (9-16).

Based on the model of classes of sufficient and component causes proposed by Rothman (17), it would be possible to consider all possible coexistence alternatives for the diseases in study, in an exploratory analysis, and thus assess their real impact on morbidity. Though, the multiple possible combinations would make the model infeasible and the

existence of infrequent combinations (for example, the coexistence of all the diseases under study) would make the model less representative of reality. Thus, in this study, a model that considers a more realistic approach to multimorbidity, based on the Rothman model of sufficient causes (17), and studies its impact on specific morbidity indicators, based on the methodology proposed by Perruccio, *et al*. (4), is proposed.

However, the use of this methodology to monitor population health will only be possible through its replication. Therefore, the programming in code of the data analysis is essential to allow its reproducibility and an efficient use of health surveys in monitoring the health of the population.

This study proposes to be the first to assess the impact of chronic diseases on specific morbidity indicators in Portugal, by NUTS II regions (level 2 of the Classification of Territorial Units for Statistics), and the first to consider the patterns of coexistence of chronic diseases in the analysis of their impact on morbidity. Also, the programming of the analysis will allow the use of the proposed methodology in similar studies (using future national health surveys, regional health surveys, national health surveys of other countries, among others), to be used for population health monitoring and to inform public health policy.

2. Objective

The objective of this study was to analyse the impact of chronic diseases coexistence on specific morbidity indicators in the Portuguese population and in the population of each region (NUTS II). As specific objectives, for the Portuguese population and the population of each region (NUTS II), we aimed to:

- characterize patterns of coexistence of chronic diseases;
- estimate associations between diseases or groups of chronic diseases and specific morbidity indicators;
- estimate the fractions of specific morbidity indicators attributable to chronic diseases or groups of diseases;
- code the data analysis performed allowing its reproducibility.

3. Methods

3.1. Participants and data collection

In order to apply this methodology, the fifth Portuguese National Health Survey (NHS) 2014 was used. This survey is the result of a partnership between Statistics Portugal, National Health Institute Dr. Ricardo Jorge and Eurostat. Its methodology is in accordance with the second wave of the European Health Interview Survey (EHIS wave 2) (18). Given the expected regularity of national health surveys and their national and regional representativeness, by NUTS II, they are an important source of data for monitoring the health of the Portuguese population.

The target population of the NHS 2014 was all individuals aged 15 years or older who resided in the Portuguese territory during the reference period (September 10 to December 15, 2014). The sampling frame was all the households on the national territory and the statistical unit of observation was the private domestic household and the selected individual. The sample size included a total of 22,538 housing units, ensuring a relative error of not more than 10% for the variables of self-reported chronic diseases and self-perceived general health by region (NUTS II). In the sample selection, a multistage stratified cluster sampling method, by NUTS II regions, was used. One individual was selected to participate per household.

Data were collected using a questionnaire, normalized according to the EHIS wave 2 methodological manual (18). The questionnaire was applied by face-to-face interview with computer or by self-completion in an electronic questionnaire via web, without possibility of proxy respondents.

3.2. Data analysis

Questions assessing health-related dimensions in the questionnaire were dichotomized to define fourteen specific morbidity indicators, that were considered as outcomes: [1] negative self-perceived general health (if “bad” or “very bad” self-perceived general health was reported), [2] absence from work (asked only to employed individuals and considered present when the participant reported 15 or more days of absence from work in the previous 12 months), [3] physical functional limitations (if the participant reported "some difficulty", "a lot of difficulty" or "unable to” walk 500m and/or 200m on level ground and/or climb up/down 12 steps, due to a chronic disease), [4] personal care activities limitations (recorded only for individuals over 65 years of age, and considered present if "some difficulty", "a lot of difficulty" or "unable to” feed himself/herself, get in/out of a bed or chair, dress/undress, use toilets, bath/shower and/or wash hands and face without help was reported), [5] household activities limitations (only for individuals over 65 years of age, and present if "some difficulty", "a lot of

difficulty" or "unable to" was reported for the following activities: prepare meals, use the telephone, shop, manage medication, do light housework, do occasional heavy housework and/or take care of finances and everyday administrative tasks without help), [6] intensity of bodily pain (if "severe" or "very severe" bodily pain during the previous four weeks was reported), [7] impact of the pain on daily life (if "quite a bit" or "extremely" interference of bodily pain with the participant normal work in the previous four weeks was referred, including both work outside the home and housework), [8] hospitalisation as inpatient (when the participant referred to be admitted as an inpatient to a hospital (for an overnight or longer stay), at least once in the previous 12 months, excluding visits to emergency departments or as outpatient only), [9] hospitalisation as day patient (when the participant had, at least one admission to hospital as a day patient, that is admitted to a hospital for diagnostic, treatment or other types of health care that do not required to remain overnight, in the previous 12 months), [10] consultation of a general practitioner (present if the participant reported one or more consultations of a general practitioner or family doctor during the previous four weeks for personal treatment), [11] consultation of other specialist (when one or more consultations of a medical or surgical specialist during the previous four weeks for personal treatment was reported), [12] use of medicines prescribed (present when the participant reported the use of any medicines prescribed by a doctor during the previous two weeks, excluding contraception), [13] use of medicines not prescribed (if the participant had used any medicines, herbal medicines or vitamins not prescribed by a doctor during the previous two weeks), and [14] out-of-pocket health expenditure (considered present if the participant had expenses higher than the median of the general population, including expenses with medical consultations, complementary diagnostic tests, medications, surgeries or other treatments, in the previous two weeks).

Seventeen chronic diseases were considered as exposure: [1] asthma (allergic asthma included), [2] chronic pulmonary disease (including chronic bronchitis, chronic obstructive pulmonary disease and emphysema), [3] myocardial infarction (chronic consequences of myocardial infarction), [4] coronary heart disease (or angina pectoris), [5] hypertension, [6] stroke (including cerebral haemorrhage, cerebral thrombosis or chronic consequences of stroke), [7] arthrosis (excluding arthritis), [8] low back/neck disorder (or other chronic back/neck defect), [9] diabetes, [10] allergy (such as rhinitis, hay fever, eye inflammation, dermatitis, food allergy or other allergy, excluding allergic asthma), [11] cirrhosis, [12] urinary incontinence (including other problems in controlling the bladder), [13] kidney problems and [14] depression. Each of these 14 diseases were considered present if participants self-reported as having them during the previous 12 months. Visual impairment [15] was considered present when "some difficulty", "a lot of difficulty" or "unable to" see even when wearing glasses or contact lenses was reported, [16] hearing impairment was present when the participant had "some difficulty",

"a lot of difficulty" or was "unable to" hear in a quiet and/or in a noisier room, even when using a hearing aid, and [17] obesity was present when the estimated body mass index from the participant was equal or above 30Kg/m^2 (calculated from self-reported weight and height).

Sex, age group, level of education and monthly net income (by quintiles) of the respondents were considered as confounding variables.

The data analysis was divided in two phases: [1] initial exploratory analysis to study the coexistence of diseases, and [2] confirmatory analysis to study the association and the impact of the chronic diseases (considering their coexistence) on the specific morbidity indicators. For each phase of the analysis only individuals with complete data for the study variables were considered.

3.2.1. Exploratory analysis

The coexistence of the chronic diseases was studied by a principal component analysis (PCA), where chronic diseases with a positive correlation (factor loadings higher than 0.40) were grouped into principal components so that each principal component was mutually exclusive in relation to the diseases. The biological plausibility for this grouping was also considered. A correlation matrix with the Pearson correlation (equivalent to the phi coefficient when applied to dichotomous variables) and varimax rotation, both adapted to dichotomous variables, were used.

Through the statistical analysis of PCA and biological interpretability, we intended to group the diseases by their natural coexistence in the population. The principal components were used as independent variables for the first level of analysis of the proposed model (Figure 1).

This exploratory analysis, as well as the grouping of the diseases based on their coexistence, allowed the recodification of the independent variables initially considered (chronic diseases) in possible sufficient classes of component causes, for each principal component of the PCA, representing the possible coexistence of chronic diseases within each component. The resulting combination of chronic diseases was used as final independent variables for the second level of the analysis of the proposed model (Figure 1).

3.2.2. Confirmatory analysis

The prevalence ratio (PR) was used as the measure of association as it is the adequate measure for cross-sectional studies when the outcome is frequent. The adjusted prevalence ratios (PR_{aj}) for the confounding variables were calculated by a multivariate log-Poisson regression model (4,19,20). For the Portuguese population and the population of each region (NUTS II), the PR_{aj} of the specific morbidity indicators (dependent variables) in chronic

diseases, considering their coexistence (independent variables), were calculated by the exponential of the Log-Poisson model coefficients. Models were adjusted to confounding variables and other components, and also to confounding variables only.

The PAF of chronic diseases, taking into account their coexistence, on specific morbidity indicators, in the Portuguese population and in the population of each region (NUTS II), were calculated using the formula described in the schematization of the proposed conceptual model (Figure 1) (4,21).

The statistical analysis took into account the sampling methodology of the NHS 2014, by applying a final weight to each statistical unit of the sample. The significance level defined for the statistical analysis was 5%.

3.2.3. Operationalization of statistical analysis

The data analysis and its programming in code was made through the programming language R, version 3.4.1, using the integrated development environment RStudio Desktop, version 1.0.153. In addition to the base packages in R, the *foreign*, *psych*, *questionr*, *shiny*, *DT*, *png*, *formattable*, *rmarkdown* and *knitr* packages were used.

The complete R code used in the data analysis of the present study is systematized in the supplementary material.

4. Results

Of the 18,204 individuals interviewed in the NHS 2014, 17,739 had complete data on the chronic diseases under study and on the confounding variables. Only these participants were included in the exploratory analysis (465 individuals with missing data were excluded).

In the confirmatory analysis, individuals with missing data regarding the morbidity indicators in analysis were excluded from the respective model. Thus, between zero (in the "use of medicines not prescribed" analysis) and 569 (in the "out-of-pocket health expenditure" analysis) individuals were excluded, totalizing 17,170 to 17,739 individuals included in each specific morbidity indicator model. It should be noted that for the morbidity indicators analysed in restricted subgroups of the population, namely the indicator "absence from work" that included only individuals employed, and the "personal care activities limitations" and "household activities limitations" indicators including individuals over 65 years of age, the total number of individuals considered in the final sample was 7,622 and 5,475, with only 37 and one individual excluded due missing data, respectively.

Of the surveyed individuals, 69.54% reported at least one of the chronic diseases under study (Table 1). The most prevalent disease was low back/neck disorder (32.62%), followed by hypertension (25.13%) and arthrosis (23.88%). On the other hand, stroke (1.89%), myocardial infarction (1.72%) and cirrhosis (0.65%) were the least reported diseases.

Regarding the specific morbidity indicators, the use of medicines prescribed (55.95%), out-of-pocket health expenditure (54.51%) and household activities limitations in individuals over 65 years of age (49.43%) were the most prevalent (Table 2). On the other hand, impact of the pain on daily life (9.63%), hospitalisation as inpatient (9.09%) and absence from work in employed individuals (8.51%) were the least reported. Only 11.98% of the individuals reported none morbidity indicator (considering the criteria and selected thresholds).

4.1. Exploratory analysis

Eight principal components were selected from the PCA for the final analysis, taking into account the defined criteria.

Although these eight components explained 62.6% of the cumulative variance, with an eigenvalue of less than one (0.896) (Table 3), and without corresponding to an inflection point of the scree plot (Figure 2), this number of principal components was selected since it represented the minimum number of components where each chronic disease corresponded to only one component (for factor loadings greater than 0.40). Additionally, the eight principal components had biological plausibility and an increase in the number of principal components did not significantly increase the explained cumulative variance.

Thus, chronic diseases under study were grouped into [1] *arthrosis, low back/neck disorder, allergy and depression*, [2] *asthma and chronic pulmonary disease*, [3] *visual impairment and hearing impairment*, [4] *hypertension, diabetes and obesity*, [5] *urinary incontinence and kidney problems*, [6] *myocardial infarction and coronary heart disease*, [7] *stroke* and [8] *cirrhosis* (Table 4).

4.2. Confirmatory analysis

In this paper, only the results for the overall Portuguese population will be briefly described, and only the graphical output (an arrow diagram comparing the hierarchy of statistically significant PR_{aj} ($p<0.05$) with their respective PAF) for the specific morbidity indicator of negative self-perceived general health will be presented. All the results and graphical and tabular outputs for Portugal and NUTS II regions populations, for all specific morbidity indicators, for both levels of analysis of the model (by principal components and by possible classes of sufficient causes), adjusted only for the confounders and for the confounders and other components, can be consulted online on the website https://morbilidade.github.io/en/.

4.2.1. Negative self-perceived general health

The grouping of diseases with the greatest impact on the negative self-perceived general health was *arthrosis, low back/neck disorder, allergy and/or depression* (61.96%). The coexistence of *arthrosis and low back/neck disorder* contributed to more than 30% of negative self-perceived general health, in Portugal (Figure 3).

4.2.2. Absence from work (only employed individuals)

Low back/neck disorder, allergy and/or depression had the greatest impact on work absenteeism (39.40%), with *low back/neck disorder* with or without *arthrosis* contributing to more than 12%.

4.2.3. Physical functional limitations

The presence of any of the diseases under study contributed to more than 90% of the physical functional limitations, and the contribution of the component of *arthrosis, low back/neck disorder, allergy and/or depression* was 56.32%. The coexistence of *arthrosis and low back/neck disorder*, with or without *depression*, contributed to more than 30% for this morbidity indicator.

4.2.4. Personal care activities limitations (in individuals over 65 years of age)

Again, in relation to the limitations in personal care activities in individuals over 65 years of age, the presence of any of the diseases under study contributed to more than 90% of this indicator. The coexistence of *visual and hearing impairment* contributed to 18.45%.

4.2.5. Household activities limitations (in individuals over 65 years of age)

Also in individuals over 65 years of age, only 56.75% of the household activities limitations was explained by the existence of the chronic diseases under study. The principal component of *arthrosis, low back/neck disorder, allergy and/or depression* was, again, the one with the greatest impact on this morbidity indicator (32.48%).

4.2.6. Intensity of bodily pain

The principal component of *arthrosis, low back/neck disorder, allergy and/or depression* contributed to the great majority of bodily pain intensity (63.50%), mainly due to *low back/neck disorder and arthrosis*.

4.2.7. Impact of the pain on daily life

The results of the impact of the pain on daily life are similar to those of the intensity of bodily pain, being the component of *arthrosis, low back/neck disorder, allergy and/or depression* the main contributor to this indicator.

4.2.8. Hospitalisation as inpatient

Only 39.38% of hospital admissions were explained by the questioned chronic diseases, and the group of *arthrosis, low back/neck disorder, allergy and/or depression* contributed to 20.54% of these.

4.2.9. Hospitalisation as day patient

The contribution of the chronic diseases under study to hospitalisation as day patient was only 23.33%, and lower than their contribution to hospitalisation as inpatient. The component of *arthrosis, low back/neck disorder, allergy and/or depression* remained the one with the greatest impact on this indicator of morbidity.

4.2.10. Consultation of a general practitioner

The main chronic diseases that contributed to a recent consultation of a general practitioner were the coexistence of *arthrosis and low back/neck disorder*, and *hypertension*

and diabetes, isolated or concomitant, although all the diseases under study only accounted for 27.68% of this indicator.

4.2.11. Consultation of other specialist

The contribution of the studied diseases to consultation of other specialist (31.44%) was slightly higher than of a general practitioner, with the greatest contribution from the coexistence of *arthrosis and low back/neck disorder*, with or without *depression*.

4.2.12. Use of medicines prescribed

Considering the combinations of diseases under study, *hypertension* was the one with the greatest impact on the consumption of medicines prescribed by a doctor (5.63%), followed by the coexistence of *arthrosis and low back/neck disorder* (2.59%).

4.2.13. Use of medicines not prescribed

Only 7.25% of the consumption of medicines not prescribed by a doctor was attributable to the presence of any of the diseases under study. However, the component of *arthrosis, low back/neck disorder, allergy and/or depression* contributed to approximately 11.42% of this morbidity indicator.

4.2.14. Out-of-pocket health expenditure

For total out-of-pocket health expenditure, the two principal components were *arthrosis, low back/neck disorder, allergy and/or depression* (11.01%) and *hypertension, diabetes and/or obesity* (5.13%), and mainly due to the coexistence of *arthrosis and low back/neck disorder* and *hypertension*.

5. Discussion

In our study, the pattern of multimorbidity in the Portuguese population in 2014 was better represented by the grouping of chronic diseases in eight components. This grouping, and the combinations of diseases within each component, allowed a more realistic assessment of the individual association and population impact of chronic diseases on the specific morbidity indicators.

The results found were in agreement with those previously reported by Perruccio *et al*. (4), among other studies (5-8), especially in relation to the differences between the association measures (PR) and the impact of the chronic diseases (PAF) on the specific morbidity indicators, and their hierarchy. Overall, the principal component with the greatest impact on the morbidity indicators under study was *arthrosis, low back/neck disorder, allergy and/or depression*. Considering the chronic diseases and their coexistence, it was also the *arthrosis and low back/neck disorder*, isolated or coexisting with other diseases, those with the greatest impact on morbidity. These results are consistent with those previously reported, where musculoskeletal diseases usually have the highest impact on the morbidity indicators considered in each study (4-8,22,23).

The differences found between the hierarchy of the individual association and the population impact, as well as the distinct hierarchies between morbidity indicators described, reinforce the importance of this type of studies and the adaptation of the respective results to the needs of the health evaluation in question (if individual or population based, and considering the appropriate morbidity indicator to be assessed). Thus, prioritization of interventions should be targeted to different chronic diseases, depending on the health-related dimension in focus and the level of action, whether it is individual (as for example in clinical practice) or population based (such as health policies, health planning, and healthcare services organization and management).

Taking into account the calculation of PAF, interventions aimed at reducing the impact of chronic diseases on morbidity, on a population level, may be directed to reduce the prevalence of the diseases, by reducing their incidence through interventions aimed at their determinants, with predictable long-term outcomes, but may also be targeted at more effective control and management of these diseases, potentially with a decrease in the association between the chronic disease and the morbidity indicator in question.

The reduced impact of having at least one disease in some of the morbidity indicators studied (including indicators on the use of healthcare services, use of medicines and out-of-pocket health expenditure) may be due to the fact that acute diseases were not considered in this study, and these diseases have an important impact on these indicators (such as acute

infectious diseases or injuries). Also, the absence, in the NHS 2014, of questions about other important chronic diseases for the study of morbidity (such as oncological diseases or other psychiatric diseases) can also explain this finding. However, in indicators that represent less specific dimensions such as negative self-perceived general health, physical functional limitations and personal care activities limitations, the existence of at least one of the chronic diseases inquired contributed to more than 90% of the PAF of each of these indicators.

Though, the fact that this study is based on data from a national health survey imposes some limitations on its interpretation. Indeed, its cross-sectional nature hampers the establishment of a causal relationship between the exposure (chronic diseases) and outcome (specific morbidity indicators) variables. However, the statistical analysis conducted, that include the adjustment of PR using a multivariate log-Poisson model to calculate the PAF, is adequate for cross-sectional population based studies (4,19,20). On the other hand, as the data collected is self-reported, its validity may be affected. This potential information bias is not limited to a possible memory bias, but also to possible misinterpretations of the questions posed or to a lack of self-awareness about personal health conditions. The lack of self-awareness about health and illness can explain, for example, the high coexistence of asthma and chronic lung disease and reduced coexistence of asthma with allergies. Nevertheless, self-reported data from surveys is a valid and mainly economic and practical measure of morbidity, even taking into account their limitations (24-27).

The methodological option for the exploratory analysis, not entirely supported in previous studies but mostly in theoretical concepts, can also be questioned. Though, the final grouping of chronic diseases by principal components was, in addition to biological plausible, consistent with previous studies, either using PCA (28,29) or other methodologies such as factor analysis (30,31), latent class analysis (29,32) or cluster analysis (29,33,34).

The strong correlation of each disease with its principal component and the reduced correlation with the other components, reinforce the validity of the grouping of chronic diseases considered. Furthermore, the selected exploratory analysis facilitates the interpretation of PAF of the multimorbidity patterns of chronic diseases, contributing to its effective translation and applicability in public health planning and policy making.

Also, the interpretation of PAF is dependent on theoretical assumptions that were not necessarily verified in the present study, namely the existence of a causal relationship between exposure and outcome, the complete reversibility of exposure (chronic diseases) and the unchanged distribution of the remaining exposures, in this case of other components/classes, upon elimination of the components/classes of interest (PAF estimates do not take into account comorbidities, the coexistence of diseases with causal relation between them) (4,21).

The option of restricting the analysis to individuals with complete information for chronic diseases and each morbidity indicator, and the chosen criteria for the classification of the morbidity indicators and their thresholds, may be also questioned. However, the low number of excluded individuals compared to the total sample size of the NHS 2014, and the choice of thresholds based on previous studies, support those methodological decisions.

The statistical significance of PR_{aj} and PAF was analysed through the calculation of p values, as the computing of confidence intervals by bootstrapping for every model would involve a huge computational cost.

This study is strengthened by the use of data from a large representative sample of the Portuguese population, and of each region (NUTS II) population, and the large number of chronic diseases (a total of 17 diseases) and specific morbidity indicators (a total of 14 indicators), that reinforce the results found.

Based on the literature review carried out, this study represents one of the first to assess the impact of numerous chronic diseases on several specific morbidity indicators and the first of its kind conducted in the population of Portugal and of its regions (NUTS II). This study is also pioneer on considering the patterns of coexistence of chronic diseases in the analysis of their impact on morbidity, through a realistic analysis of the distribution of multimorbidities in the population.

Also, and of particular relevance to view and interpret the results, the produced outputs allow a rapid comparison between individual association (PR) and population impact (PAF) of the studied chronic diseases on each of the specific morbidity indicators. Furthermore, the programming in code of the analysis and outputs will allow their reproducibility, not only for validation purposes, but also their use and adaptation to similar studies (future national health surveys, regional health surveys, national health surveys of other countries, among others), with the final objective of monitor the health of populations.

6. Conclusions

In this study, we observed that musculoskeletal diseases, such as *low back/neck disorder and arthrosis*, were the chronic diseases that had the greatest impact on the morbidity of the Portuguese population in 2014.

Depending on the morbidity indicator studied and whether the analysis is focused on individual association or population impact, different hierarchies of chronic diseases and grouping of diseases can be found. This needs to be taken into account in health planning, as different chronic diseases or coexistence of diseases should be defined as priority targets for intervention, depending on which health dimension is being considered, and whether the focus is individual healthcare services or public health policies and initiatives addressed to a population.

In addition to the importance of studying the impact of chronic diseases on morbidity, to increase the knowledge of the health and well-being of populations, as an essential complement for evidence based health planning and healthcare services management, the programming of the analysis conducted will ultimately allow its efficient use in population health monitoring.

7. References

1. Observatório Nacional das Doenças Reumáticas. Doenças Reumáticas em Portugal: da Investigação às Políticas de Saúde. 1st ed. Porto, Portugal: Instituto de Saúde Pública da Universidade do Porto; 2014. ISBN: 978-989-98867-0-4.
2. Machado V, Lima G, Teixeira C, Felício MM. Global Burden of Disease in the Northern Region of Portugal [Internet]. Porto, Portugal: Public Health Departement of the Northern Region Health Administration, Public Institute; 2011 April. Avaiable at http://portal.arsnorte.min-saude.pt/portal/page/portal/ARSNorte/Conte%C3%BAdos/Sa%C3%BAde%20P%C3%BAblica%20Conteudos/Carga_Global_Doenca_Regiao_Norte_2004.pdf.
3. GBD 2015 DALYs and HALE Collaborators. Global, regional, and national disability-adjusted life-years (DALYs) for 315 diseases and injuries and healthy life expectancy (HALE), 1990-2015: a systematic analysis for the Global Burden of Disease Study 2015. Lancet. 2016 Oct 8;388(10053):1603-1658. doi: 10.1016/S0140-6736(16)31460-X.
4. Perruccio AV, Power JD, Badley EM. The relative impact of 13 chronic conditions across three different outcomes. J Epidemiol Community Health. 2007 Dec;61(12): 1056-61.
5. Griffith L, Raina P, Wu H, Zhu B, Stathokostas L. Population attributable risk for functional disability associated with chronic conditions in Canadian older adults. Age Ageing. 2010 Nov;39(6):738-45. doi: 10.1093/ageing/afq105. Epub 2010 Sep 1.
6. Slater M, Perruccio AV, Badley EM. Musculoskeletal comorbidities in cardiovascular disease, diabetes and respiratory disease: the impact on activity limitations; a representative population-based study. BMC Public Health. 2011 Feb 3;11:77. doi: 10.1186/1471-2458-11-77.
7. Palazzo C, Ravaud JF, Trinquart L, Dalichampt M, Ravaud P, Poiraudeau S. Respective contribution of chronic conditions to disability in France: results from the national Disability-Health Survey. PLoS One. 2012;7(9):e44994. Epub 2012 Sep 14.
8. Molarius A, Janson S. Self-rated health, chronic diseases, and symptoms among middle-aged and elderly men and women. J Clin Epidemiol. 2002 Apr;55(4):364-70.
9. Menotti A, Mulder I, Nissinen A, Giampaoli S, Feskens EJM, Kromhout D. Prevalence of morbidity and multimorbidity in elderly male populations and their impact on 10-year all-cause mortality: The FINE study (Finland, Italy, Netherlands, Elderly). J Clin Epidemiol. 2001;54(7):680-6.

10. Fortin M, Lapointe L, Hudon C, Vanasse A, Ntetu AL, Maltais D. Multimorbidity and quality of life in primary care: a systematic review. Health Qual Life Outcomes. 2004 Sep 20;2:51.
11. Wolff JL, Starfield B, Anderson G. Prevalence, expenditures, and complications of multiple chronic conditions in the elderly. Arch Intern Med. 2002 Nov 11;162(20): 2269-76.
12. Glynn LG, Valderas JM, Healy P, Burke E, Newell J, Gillespie P, et al. The prevalence of multimorbidity in primary care and its effect on health care utilization and cost. Fam Pract. 2011 Oct;28(5):516-23. doi: 10.1093/fampra/cmr013. Epub 2011 Mar 24.
13. Violan C, Foguet-Boreu Q, Flores-Mateo G, Salisbury C, Blom J, Freitag M, et al. Prevalence, determinants and patterns of multimorbidity in primary care: a systematic review of observational studies. PLoS One. 2014 Jul 21;9(7):e102149. doi: 10.1371/journal.pone.0102149. eCollection 2014.
14. Moffat K, Mercer SW. Challenges of managing people with multimorbidity in today's healthcare systems. BMC Fam Pract. 2015 Oct 14;16:129. doi: 10.1186/s12875-015-0344-4.
15. Goodman RA, Posner SF, Huang ES, Parekh AK, Koh HK. Defining and measuring chronic conditions: imperatives for research, policy, program, and practice. Prev Chronic Dis. 2013;10:E66.
16. Prados-Torres A, Calderón-Larrañaga A, Hancco-Saavedra J, Poblador-Plou B, van den Akker M. Multimorbidity patterns: a systematic review. J Clin Epidemiol. 2014 Mar;67(3):254-66. doi: 10.1016/j.jclinepi.2013.09.021.
17. Rothman KJ, Greenland S, Lash TL, editors. Modern epidemiology. 3., [rev. and updated] ed. Philadelphia, Pa.: Wolters Kluwer, Lippincott Williams & Wilkins; 2008. ISBN: 978-1451190052.
18. European Commission, Eurostat. European Health Interview Survey (EHIS wave 2) methodological manual: 2013 edition. [Internet]. Luxembourg: Publications Office; 2013. Disponível em: http://dx.publications.europa.eu/10.2785/43280.
19. Coutinho LM, Scazufca M, Menezes PR. Methods for estimating prevalence ratios in cross-sectional studies. Rev Saude Publica. 2008 Dec;42(6):992-8.
20. Barros AJ, Hirakata VN. Alternatives for logistic regression in cross-sectional studies: an empirical comparison of models that directly estimate the prevalence ratio. BMC Med Res Methodol. 2003 Oct 20;3:21.
21. Rockhill B, Newman B, Weinberg C. Use and misuse of population attributable fractions. Am J Public Health. 1998 Jan;88(1):15-9.

22. Badley EM, Rasooly I, Webster GK. Relative importance of musculoskeletal disorders as a cause of chronic health problems, disability, and health care utilization: findings from the 1990 Ontario Health Survey. J Rheumatol. 1994 Mar;21(3):505-14.
23. Andrianakos AA, Miyakis S, Trontzas P, Kaziolas G, Christoyannis F, Karamitsos D, et al. The burden of the rheumatic diseases in the general adult population of Greece: the ESORDIG study. Rheumatology (Oxford). 2005 Jul;44(7):932-8. Epub 2005 Apr 19.
24. Beckett M, Weinstein M, Goldman N, et al. Do health interview surveys yield reliable data on chronic illness among older respondents? Am J Epidemiol 2000;151: 315–23.
25. Haapanen N, Miilunpalo S, Pasanen M, Oja P, Vuori I. Agreement between questionnaire data and medical records of chronic diseases in middle-aged and elderly Finnish men and women. Am J Epidemiol. 1997 Apr 15;145(8):762-9.
26. Heliövaara M1, Aromaa A, Klaukka T, Knekt P, Joukamaa M, Impivaara O. Reliability and validity of interview data on chronic diseases. The Mini-Finland Health Survey. J Clin Epidemiol. 1993 Feb;46(2):181-91.
27. Violán C, Foguet-Boreu Q, Hermosilla-Pérez E, Valderas JM, Bolíbar B, Fàbregas-Escurriola M, Brugulat-Guiteras P, Muñoz-Pérez MÁ. Comparison of the information provided by electronic health records data and a population health survey to estimate prevalence of selected health conditions and multimorbidity. BMC Public Health. 2013 Mar 21;13:251. doi: 10.1186/1471-2458-13-251.
28. Holden L, Scuffham PA, Hilton MF, Muspratt A, Ng SK, Whiteford HA. Patterns of multimorbidity in working Australians. Popul Health Metr. 2011 Jun 2;9(1):15. doi: 10.1186/1478-7954-9-15.
29. Islam MM, Valderas JM, Yen L, Dawda P, Jowsey T, McRae IS. Multimorbidity and comorbidity of chronic diseases among the senior Australians: prevalence and patterns. PLoS One. 2014 Jan 8;9(1):e83783. doi: 10.1371/journal.pone.0083783. eCollection 2014.
30. Prados-Torres A, Poblador-Plou B, Calderon-Larranaga A, Gimeno-Feliu LA, Gonzalez-Rubio F, Poncel-Falco A, Sicras-Mainar A, Alcalá-Nalvaiz JT. Multimorbidity Patterns in Primary Care: Interactions among Chronic Diseases Using Factor Analysis. PLoS One. 2012;7(2):e32190. doi: 10.1371/journal.pone. 0032190. Epub 2012 Feb 29.
31. Schafer I, von Leitner EC, Schon G, Koller D, Hansen H, Kolonko T, et al. Multimorbidity Patterns in the Elderly: A New Approach of Disease Clustering

Identifies Complex Interrelations between Chronic Conditions. PLoS One. 2010 Dec 29;5(12):e15941. doi: 10.1371/journal.pone.0015941.

32. Simões D, Araújo FA, Severo M, Monjardino T, Cruz I, Carmona L, Lucas R. Patterns and Consequences of Multimorbidity in the General Population: There is No Chronic Disease Management Without Rheumatic Disease Management. Arthritis Care Res (Hoboken). 2017 Jan;69(1):12-20. doi: 10.1002/acr.22996. Epub 2016 Nov 17.
33. Marengoni A, Rizzuto D, Wang HX, Winblad B, Fratiglioni L. Patterns of chronic multimorbidity in the elderly population. J Am Geriatr Soc. 2009 Feb;57(2):225-30. doi: 10.1111/j.1532-5415.2008.02109.x.
34. John R, Kerby DS, Hennessy CH. Patterns and impact of comorbidity and multimorbidity among community-resident American Indian elders. Gerontologist. 2003 Oct;43(5):649-60.

8. Figures

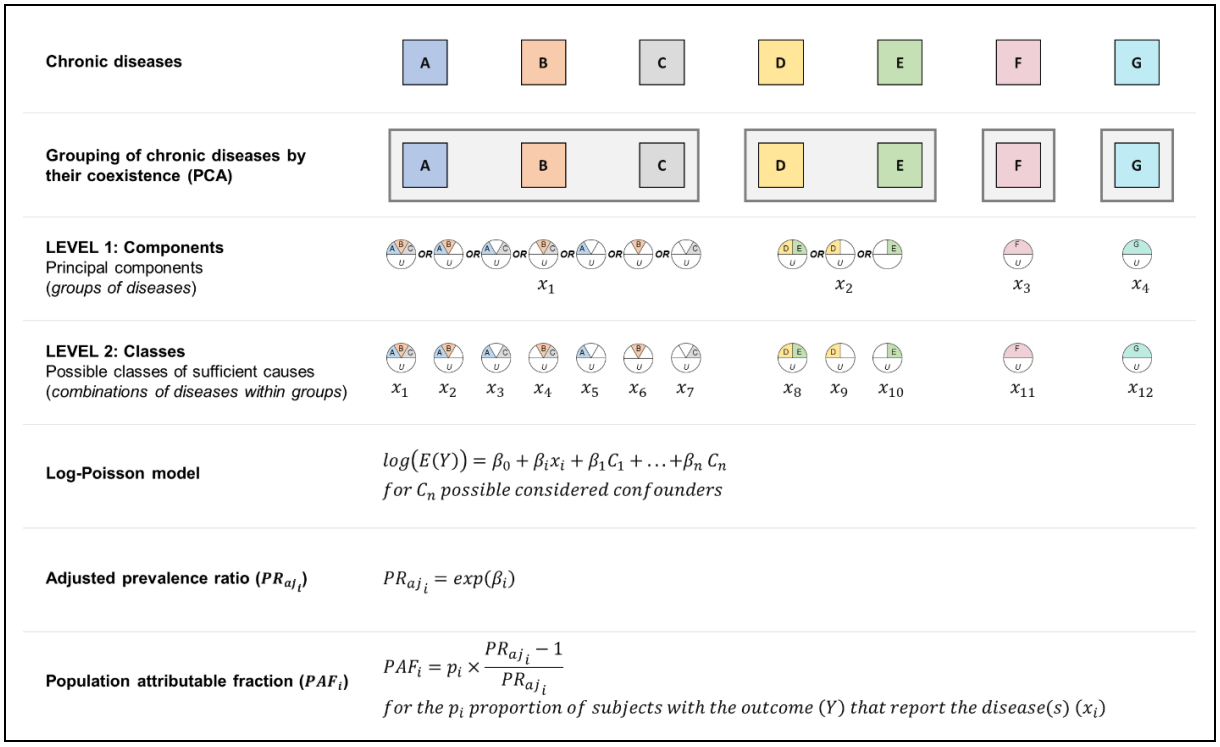


Figure 1. Conceptual model exemplified with seven diseases and four principal components (only for demonstration purposes)

Legend: *PAF* Population attributable fraction; *PCA* Principal component analysis; *PRaj* Adjusted prevalence ratio.

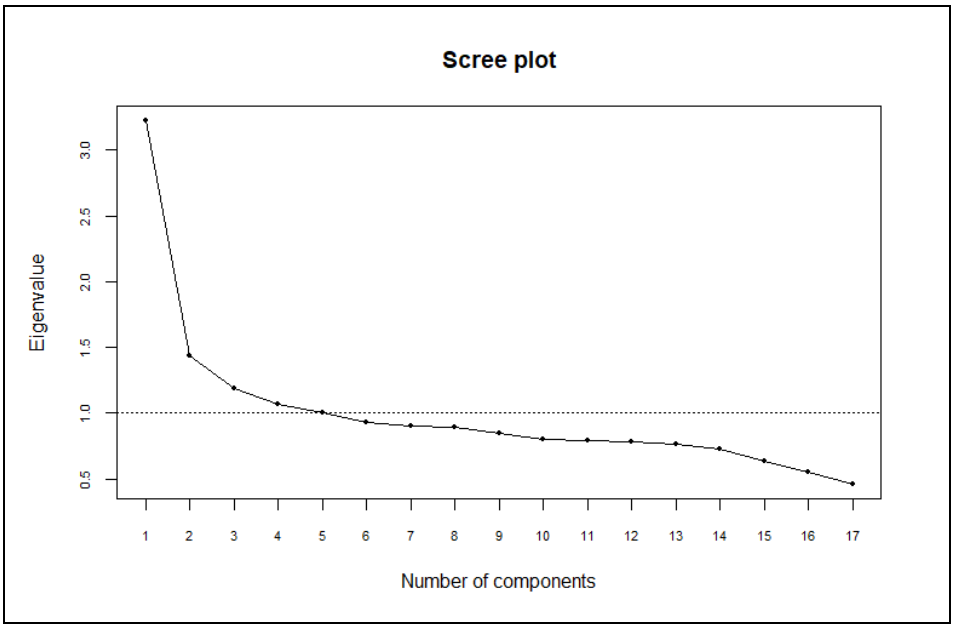


Figure 2. Scree plot of the principal component analysis (PCA)

Legend: *PCA* Principal component analysis.

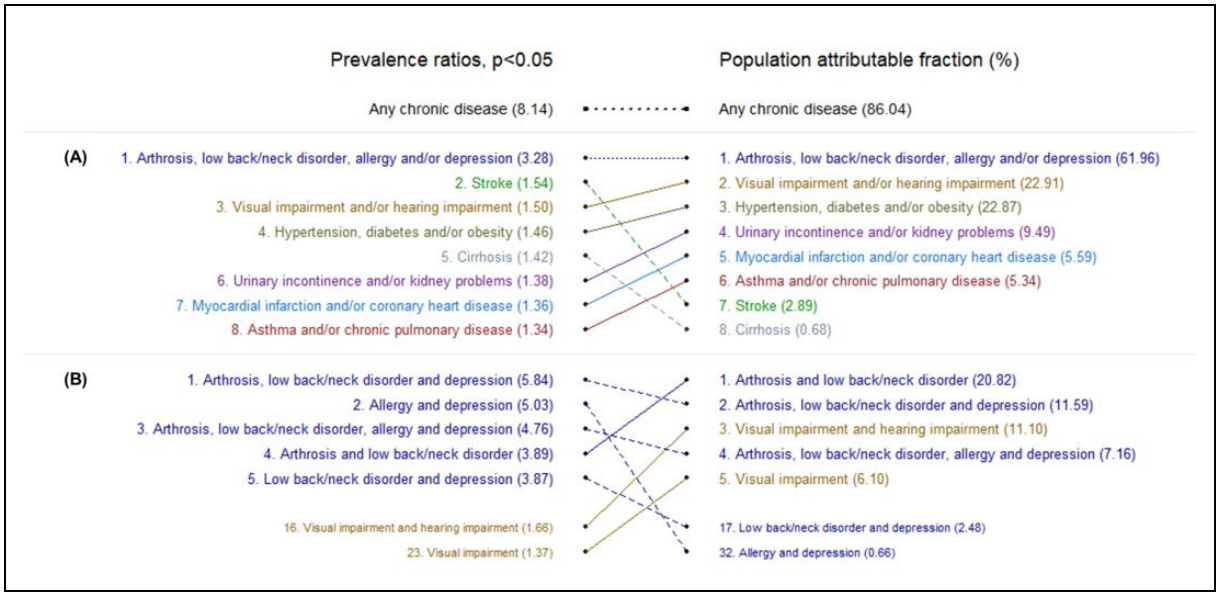


Figure 3. Negative self-perceived general health in Portugal for all eight principal components (A) and for the five main classes (B), adjusted for confounding variables and other components

9. Tables

Table 1. Prevalence of the chronic diseases, in Portugal

	Total	n	Prevalence (%)	Weighted prevalence (%)
Any chronic disease	17739	13012	73.35	69.54
Low back/neck disorder	17739	6427	36.23	32.62
Hypertension	17739	5294	29.84	25.13
Arthrosis	17739	5027	28.34	23.88
Visual impairment	17739	4636	26.13	23.00
Hearing impairment	17739	3987	22.48	20.48
Allergy	17739	3360	18.94	19.40
Obesity	17739	3187	17.97	16.22
Depression	17739	2363	13.32	11.83
Diabetes	17739	2000	11.27	9.29
Urinary incontinence	17739	1458	8.22	7.19
Chronic pulmonary disease	17739	1155	6.51	5.77
Asthma	17739	980	5.52	5.03
Kidney problems	17739	1038	5.85	4.53
Coronary heart disease	17739	958	5.40	4.22
Stroke	17739	419	2.36	1.89
Myocardial infarction	17739	380	2.14	1.72
Cirrhosis	17739	124	0.70	0.65

Legend: *n* Number of individuals with the chronic disease, in descending order of weighted prevalence.

Table 2. Prevalence of the specific morbidity indicators, in Portugal

	Total	n	Prevalence (%)	Weighted prevalence (%)
Use of medicines prescribed	17737	10668	60.15	55.95
Out-of-pocket health expenditure	17170	8622	50.22	54.51
Household activities limitations (>65 years old)	5475	2839	51.85	49.43
Hospitalisation as day patient	17733	7327	41.32	40.41
Consultation of a general practitioner	17715	4427	24.99	25.05
Use of medicines not prescribed	17739	4326	24.39	23.88
Personal care activities limitations (>65 years old)	5475	1128	20.60	20.99
Consultation of other specialist	17726	2697	15.21	16.09
Intensity of bodily pain	17729	2868	16.18	15.26
Negative self-perceived general health	17735	2858	16.12	13.16
Physical functional limitations	17714	2435	13.75	10.04
Impact of the pain on daily life	17737	1980	11.16	9.63
Hospitalisation as inpatient	17737	1656	9.34	9.09
Absence from work (only employed individuals)	7622	658	8.63	8.51

Legend: *n* Number of individuals reporting the morbidity indicator, in descending order of weighted prevalence.

Table 3. Summary of the principal component analysis (PCA)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Factor loadings	1.655	1.500	1.485	1.467	1.276	1.221	1.043	1.000
Proportion of variance	0.097	0.088	0.087	0.086	0.075	0.072	0.061	0.059
Cumulative proportion of variance	0.097	0.186	0.273	0.359	0.434	0.506	0.568	0.626
Proportion of variance explained	0.155	0.141	0.139	0.138	0.120	0.115	0.098	0.094
Cumulative proportion of variance explained	0.155	0.296	0.436	0.574	0.693	0.808	0.906	1.000

Legend: *PC* Principal component; *PCA* Principal component analysis.

Table 4. Factor loadings for each chronic disease by principal component (PC)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Arthrosis	**0.531**	0.004	0.390	0.342	0.157	0.111	-0.012	-0.020
Low back/neck disorder	**0.631**	0.010	0.337	0.245	0.091	0.090	-0.050	-0.008
Allergy	**0.619**	0.378	-0.187	-0.135	-0.085	-0.021	0.013	0.040
Depression	**0.644**	-0.017	0.071	-0.008	0.187	0.004	0.074	0.037
Asthma	0.054	**0.840**	0.024	0.044	0.024	-0.003	-0.004	-0.025
Chronic pulmonary disease	0.054	**0.793**	0.129	0.072	0.096	0.086	0.018	0.029
Visual impairment	0.066	0.068	**0.738**	0.029	0.053	-0.001	0.036	0.047
Hearing impairment	0.106	0.051	**0.693**	0.028	0.107	0.086	0.019	-0.006
Hypertension	0.331	-0.002	0.270	**0.517**	0.047	0.171	0.113	-0.012
Diabetes	0.009	0.043	0.089	**0.610**	0.147	0.025	0.220	0.091
Obesity	0.003	0.064	-0.091	**0.770**	0.008	0.001	-0.154	-0.036
Urinary incontinence	0.149	0.037	0.193	0.139	**0.682**	-0.004	0.104	-0.022
Kidney problems	0.089	0.071	0.019	0.038	**0.799**	0.130	-0.013	0.061
Myocardial infarction	-0.019	0.050	0.036	0.064	-0.053	**0.728**	0.307	0.026
Coronary heart disease	0.109	0.036	0.083	0.046	0.198	**0.775**	-0.151	0.005
Stroke	0.055	0.009	0.043	0.062	0.076	0.084	**0.905**	-0.014
Cirrhosis	0.044	0.005	0.038	0.037	0.041	0.026	-0.010	**0.989**

Legend: *PC* Principal component.

10. Supplementary material

Supplementary material

This document intends to systematize the code, in programming language *R*, used for the data analysis and output production of the article **"Monitoring the morbidity associated with chronic diseases: Study of the coexistence of chronic diseases and their impact on specific indicators of morbidity in the National Survey of Health 2014"**, published in the *Journal of Clinical Epidemiology*, and whose results can be found at *https://morbilidade.github.io/en/*.

The presented code was tested using version *R-3.4.1 for Windows*, *RStudio Desktop 1.0.153 for Windows Vista/7/8/10* and the updates to the used packages as of September 10, 2017.

The writing of this document was done through the packages `rmarkdown` and `knitr`, in *R*, using the integrated development environment *RStudio*.

The created functions are divided into the following chapters:

- "Data extraction and recoding", with the code for extraction of the data of interest from the original database, and recoding and organization for the subsequent analyzes;
- "Exploratory analysis", with the code for the principal component analysis, the outputs to decide the number of components to be used and the distribution of the individuals by the possible classes of sufficient causes from the number of selected components, according to the proposed methodology;
- "Confirmatory analysis", with the code for the calculation of the described statistics in the proposed methodology, including the prevalence ratios and population attributable fractions;
- "Outputs", with the code for the created *tool* using the `shiny` package, as well as other outputs presented in the paper.

Data extraction and recoding

The extraction and recoding of the data of interest for the study, from the original database of the Portuguese National Health Survey (NHS, INS in Portuguese) 2014 (the version used was in `.sav` format of March 8, 2016), was systematized in the function `fun01_readandrecodefun`. This function has as dependency the `foreign` package.

It has as arguments:

Arguments	Default	Description
`datafile`	"INS2014_anonimizada_20160308.sav"	Name of the file with the original NHS 2014 database, with its file extension (between quotation marks, eg: "INS2014_anonimized_20160308.sav")
`workingdirectory`	getwd()	Name of the folder where the `datafile` argument file is located (between quotation marks and with the `\\` or `/` symbol separating each level, eg: "C:\INS2014")

It has as outputs:

- *list* containing two *lists*, `data` and `labels`, each containing five *data frames*:
  - `data$diseases`: independent variables (diseases),
  - `data$outcomes`: dependent variables (morbidity indicators),
  - `data$conf`: confounding variables,
  - `data$others`: other variables of interest for the analysis (such as weights and regions),
  - `data$excl`: individuals excluded from the analysis (due to missing data on one or more diseases),
  - `labels$diseases_labels`: description of variables in `data $ diseases`,
  - `labels$outcomes_labels`: description of variables in `data$outcomes`,

  - `labels$conf_labels`: description of variables in `data$conf`,
  - `labels$others_labels`: description of variables in `data$others`,
  - `labels$excl_labels`: description of variables in `data$excl`;
- ten `.csv` files with the information contained in each of the previous *data frames*, saved in the folder passed to the `workingdirectory` argument.

```
fun01_readandrecodefun <- function(datafile = "INS2014_anonimizada_20160308.sav",
                                   workingdirectory = getwd()) {


  #### Function for reading and recoding the original database (".sav" or ".dta")


  # Open the appropriate working directory (where the ".sav" or ".dta" file is located)

  setwd(workingdirectory)


  # If necessary install the foreign package and load it

  packages <- "foreign"
  newpackages <- packages[!(packages %in% installed.packages()[,"Package"])]
  if(length(newpackages)) install.packages(newpackages)
  library(foreign)


  # Read the .sav file

  data <- read.spss(file = datafile,
                    use.value.labels = FALSE,
                    to.data.frame = TRUE)
  # If in a ".dta" file replace with:
  # data <- read.dta(file = datafile, convert.factors = FALSE)


  # Extraction of the variables of interest, divided into five temporary data frames:
  #  - intermed_orig - intermediate variables
  #  - diseases_orig - independent variables (diseases)
  #  - outcomes_orig - dependent variables (morbidity indicators)
  #  - conf_orig - confounding variables
  #  - others_orig - other variables of interest for the analysis (such as weights and regions)

  intermed_orig <- data[, c("PID", "AW1", "AW2_COD", "PL6", "IN9", "PL7", "IN14", "PC1A",
                            "PC1B", "PC1C", "PC1D", "PC1E", "IN15", "HA1A", "HA1B", "HA1C",
                            "HA1D", "HA1E", "HA1F", "HA1G", "IN55", "IN56", "IN57", "IN58",
                            "IN59", "CD1H", "CD1I", "PL4_COD", "PL5", "BM1", "BM2",
                            "MAINSTAT")]

  diseases_orig <- data[, c("PID", "CD1A", "CD1B", "CD1C", "CD1D", "CD1E", "CD1F", "CD1G",
                            "CD1J", "CD1K", "CD1L", "CD1M", "CD1N", "CD1O", "PL2_COD")]

  outcomes_orig <- data[, c("PID", "HS1", "PN1", "PN2", "HO1", "HO3", "AM3", "AM5", "MD1",
                            "MD2")]

  conf_orig <- data[, c("PID", "SEX", "AGE_COD", "HATLEVEL_COD", "HHINCOME")]
```

```
others_orig <- data[, c("PID", "WGT", "REGION")]

rm(packages, newpackages, data)
gc()


# Recoding the original variables

intermed <- intermed_orig
intermed[, -1] <- NA
intermed$AW1 <- ifelse(intermed_orig$AW1 == 2,
                       0,
                       ifelse(intermed_orig$AW1 == 1,
                              1,
                              NA))
intermed$AW2_COD <- ifelse(intermed_orig$AW2_COD == 1 |
                             intermed_orig$AW2_COD == 2,
                           0,
                           ifelse(intermed_orig$AW2_COD >= 3,
                                  1,
                                  NA))
intermed$PL6 <- ifelse(intermed_orig$PL6 == 1,
                       0,
                       ifelse(intermed_orig$PL6 >= 2,
                              1,
                              NA))
intermed$IN9 <- ifelse(intermed_orig$IN9 == 1,
                       0,
                       ifelse(intermed_orig$IN9 >= 2,
                              1,
                              NA))
intermed$PL7 <- ifelse(intermed_orig$PL7 == 1,
                       0,
                       ifelse(intermed_orig$PL7 >= 2,
                              1,
                              NA))
intermed$IN14 <- ifelse((intermed_orig$IN14 >= 1 &
                           intermed_orig$IN14 <= 7) |
                          intermed_orig$IN14 == 10|
                          intermed_orig$IN14 == -2,
                        0,
                        ifelse(intermed_orig$IN14 == 8 |
                                 intermed_orig$IN14 == 9,
                               1,
                               NA))
intermed$PC1A <- ifelse(intermed_orig$PC1A == 1,
                        0,
                        ifelse(intermed_orig$PC1A >= 2,
                               1,
                               NA))
intermed$PC1B <- ifelse(intermed_orig$PC1B == 1,
                        0,
                        ifelse(intermed_orig$PC1B >= 2,
                               1,
                               NA))
intermed$PC1C <- ifelse(intermed_orig$PC1C == 1,
```

```
                        0,
                        ifelse(intermed_orig$PC1C >= 2,
                               1,
                               NA))
intermed$PC1D <- ifelse(intermed_orig$PC1D == 1,
                        0,
                        ifelse(intermed_orig$PC1D >= 2,
                               1,
                               NA))
intermed$PC1E <- ifelse(intermed_orig$PC1E == 1,
                        0,
                        ifelse(intermed_orig$PC1E >= 2,
                               1,
                               NA))
intermed$IN15 <- ifelse(intermed_orig$IN15 == 1,
                        0,
                        ifelse(intermed_orig$IN15 >= 2,
                               1,
                               NA))
intermed$HA1A <- ifelse(intermed_orig$HA1A == 1 |
                          intermed_orig$HA1A == 5,
                        0,
                        ifelse(intermed_orig$HA1A >= 2 &
                                 intermed_orig$HA1A <= 4,
                               1,
                               NA))
intermed$HA1B <- ifelse(intermed_orig$HA1B == 1 |
                          intermed_orig$HA1B == 5,
                        0,
                        ifelse(intermed_orig$HA1B >= 2 &
                                 intermed_orig$HA1B <= 4,
                               1,
                               NA))
intermed$HA1C <- ifelse(intermed_orig$HA1C == 1 |
                          intermed_orig$HA1C == 5,
                        0,
                        ifelse(intermed_orig$HA1C >= 2 &
                                 intermed_orig$HA1C <= 4,
                               1,
                               NA))
intermed$HA1D <- ifelse(intermed_orig$HA1D == 1 |
                          intermed_orig$HA1D == 5,
                        0,
                        ifelse(intermed_orig$HA1D >= 2 &
                                 intermed_orig$HA1D <= 4,
                               1,
                               NA))
intermed$HA1E <- ifelse(intermed_orig$HA1E == 1 |
                          intermed_orig$HA1E == 5,
                        0,
                        ifelse(intermed_orig$HA1E >= 2 &
                                 intermed_orig$HA1E <= 4,
                               1,
                               NA))
intermed$HA1F <- ifelse(intermed_orig$HA1F == 1 |
                          intermed_orig$HA1F == 5,
```

```
                        0,
                        ifelse(intermed_orig$HA1F >= 2 &
                                 intermed_orig$HA1F <= 4,
                               1,
                               NA))
intermed$HA1G <- ifelse(intermed_orig$HA1G == 1 |
                          intermed_orig$HA1G == 5,
                        0,
                        ifelse(intermed_orig$HA1G >= 2 &
                                 intermed_orig$HA1G <= 4,
                               1,
                               NA))
intermed$IN55 <- ifelse(intermed_orig$IN55 == -1,
                        NA,
                        ifelse(intermed_orig$IN55 == -2,
                               0,
                               intermed_orig$IN55))
intermed$IN56 <- ifelse(intermed_orig$IN56 == -1,
                        NA,
                        ifelse(intermed_orig$IN56 == -2,
                               0,
                               intermed_orig$IN56))
intermed$IN57 <- ifelse(intermed_orig$IN57 == -1,
                        NA,
                        ifelse(intermed_orig$IN57 == -2,
                               0,
                               intermed_orig$IN57))
intermed$IN58 <- ifelse(intermed_orig$IN58 == -1,
                        NA,
                        ifelse(intermed_orig$IN58 == -2,
                               0,
                               intermed_orig$IN58))
intermed$IN59 <- ifelse(intermed_orig$IN59 == -1,
                        NA,
                        ifelse(intermed_orig$IN59 == -2,
                               0,
                               intermed_orig$IN59))
intermed$CD1H <- ifelse(intermed_orig$CD1H == 2,
                        0,
                        ifelse(intermed_orig$CD1H == 1,
                               1,
                               NA))
intermed$CD1I <- ifelse(intermed_orig$CD1I == 2,
                        0,
                        ifelse(intermed_orig$CD1I == 1,
                               1,
                               NA))
intermed$PL4_COD <- ifelse(intermed_orig$PL4_COD == 1,
                           0,
                           ifelse(intermed_orig$PL4_COD >= 2,
                                  1,
                                  NA))
intermed$PL5 <- ifelse(intermed_orig$PL5 == 1,
                       0,
                       ifelse(intermed_orig$PL5 >= 2,
                              1,
```

```
                              NA))
intermed$BM1 <- ifelse(intermed_orig$BM1 == -1,
                       NA,
                       intermed_orig$BM1)
intermed$BM2 <- ifelse(intermed_orig$BM2 == -1,
                       NA,
                       intermed_orig$BM2)
intermed$MAINSTAT <- ifelse(intermed_orig$MAINSTAT == -1,
                            NA,
                            intermed_orig$MAINSTAT)

diseases <- diseases_orig
diseases[, -1] <- NA
diseases$CD1A <- ifelse(diseases_orig$CD1A == 2,
                        0,
                        ifelse(diseases_orig$CD1A == 1,
                               1,
                               NA))
diseases$CD1B <- ifelse(diseases_orig$CD1B == 2,
                        0,
                        ifelse(diseases_orig$CD1B == 1,
                               1,
                               NA))
diseases$CD1C <- ifelse(diseases_orig$CD1C == 2,
                        0,
                        ifelse(diseases_orig$CD1C == 1,
                               1,
                               NA))
diseases$CD1D <- ifelse(diseases_orig$CD1D == 2,
                        0,
                        ifelse(diseases_orig$CD1D == 1,
                               1,
                               NA))
diseases$CD1E <- ifelse(diseases_orig$CD1E == 2,
                        0,
                        ifelse(diseases_orig$CD1E == 1,
                               1,
                               NA))
diseases$CD1F <- ifelse(diseases_orig$CD1F == 2,
                        0,
                        ifelse(diseases_orig$CD1F == 1,
                               1,
                               NA))
diseases$CD1G <- ifelse(diseases_orig$CD1G == 2,
                        0,
                        ifelse(diseases_orig$CD1G == 1,
                               1,
                               NA))
diseases$CD1J <- ifelse(diseases_orig$CD1J == 2,
                        0,
                        ifelse(diseases_orig$CD1J == 1,
                               1,
                               NA))
diseases$CD1K <- ifelse(diseases_orig$CD1K == 2,
                        0,
                        ifelse(diseases_orig$CD1K == 1,
```

```
                                1,
                                NA))
diseases$CD1L <- ifelse(diseases_orig$CD1L == 2,
                        0,
                        ifelse(diseases_orig$CD1L == 1,
                                1,
                                NA))
diseases$CD1M <- ifelse(diseases_orig$CD1M == 2,
                        0,
                        ifelse(diseases_orig$CD1M == 1,
                                1,
                                NA))
diseases$CD1N <- ifelse(diseases_orig$CD1N == 2,
                        0,
                        ifelse(diseases_orig$CD1N == 1,
                                1,
                                NA))
diseases$CD1O <- ifelse(diseases_orig$CD1O == 2,
                        0,
                        ifelse(diseases_orig$CD1O == 1,
                                1,
                                NA))
diseases$PL2_COD <- ifelse(diseases_orig$PL2_COD == 1,
                           0,
                           ifelse(diseases_orig$PL2_COD >= 2,
                                  1,
                                  NA))

outcomes <- outcomes_orig
outcomes[, -1] <- NA
outcomes$HS1 <- ifelse(outcomes_orig$HS1 >= 1 &
                         outcomes_orig$HS1 <= 3,
                       0,
                       ifelse(outcomes_orig$HS1 == 4 |
                                outcomes_orig$HS1 == 5,
                              1,
                              NA))
outcomes$PN1 <- ifelse(outcomes_orig$PN1 >= 1 &
                         outcomes_orig$PN1 <= 4,
                       0,
                       ifelse(outcomes_orig$PN1 == 5 |
                                outcomes_orig$PN1 == 6,
                              1,
                              NA))
outcomes$PN2 <- ifelse(outcomes_orig$PN2 >= 1 &
                         outcomes_orig$PN2 <= 3,
                       0,
                       ifelse(outcomes_orig$PN2 == 4 |
                                outcomes_orig$PN2 == 5,
                              1,
                              NA))
outcomes$HO1 <- ifelse(outcomes_orig$HO1 == 2,
                       0,
                       ifelse(outcomes_orig$HO1==1,
                              1,
                              NA))
```

```r
outcomes$HO3 <- ifelse(outcomes_orig$HO3 == 2,
                       0,
                       ifelse(outcomes_orig$HO3 == 1,
                              1,
                              NA))
outcomes$AM3 <- ifelse(outcomes_orig$AM3 == 0 |
                         outcomes_orig$AM3 == -2,
                       0,
                       ifelse(outcomes_orig$AM3 >= 1,
                              1,
                              NA))
outcomes$AM5 <- ifelse(outcomes_orig$AM5 == 0 |
                         outcomes_orig$AM5 == -2,
                       0,
                       ifelse(outcomes_orig$AM5 >= 1,
                              1,
                              NA))
outcomes$MD1 <- ifelse(outcomes_orig$MD1 == 2,
                       0,
                       ifelse(outcomes_orig$MD1 == 1,
                              1,
                              NA))
outcomes$MD2 <- ifelse(outcomes_orig$MD2 == 2,
                       0,
                       ifelse(outcomes_orig$MD2 == 1,
                              1,
                              NA))

conf <- conf_orig
conf[, -1] <- NA
conf$SEX <- as.numeric(conf_orig$SEX)
conf$AGE_COD <- as.numeric(conf_orig$AGE_COD)
conf$HATLEVEL_COD <- as.numeric(conf_orig$HATLEVEL_COD)
conf$HHINCOME <- as.numeric(conf_orig$HHINCOME)

others <- others_orig
others[,-1] <- NA
others$WGT <- others_orig$WGT
others$REGION <- others_orig$REGION

levels(others$REGION) <- c("North", "Algarve", "Center", "Lisbon", "Alentejo",
                           "Azores", "Madeira")

rm(intermed_orig, diseases_orig, outcomes_orig, conf_orig, others_orig)
gc()


# Creation of new variables

intermed$IN60_NEW <- intermed$IN55 + intermed$IN56 + intermed$IN57 + intermed$IN58 +
  intermed$IN59
intermed$BM3_NEW <- intermed$BM2 / (intermed$BM1/100) ^ 2

diseases$CD1HI_NEW <- ifelse(intermed$CD1H == 0 &
                               intermed$CD1H == 0,
                             0,
```

```
                              ifelse(intermed$CD1H == 1 |
                                       intermed$CD1H == 1,
                                     1,
                                     NA))
diseases$PL6_NEW <- ifelse(intermed$PL4_COD == 0 &
                             intermed$PL5 == 0,
                           0,
                           ifelse(intermed$PL4_COD == 1 |
                                    intermed$PL5 == 1,
                                  1,
                                  NA))
diseases$BM4_NEW <- ifelse(intermed$BM3_NEW < 30,
                           0,
                           ifelse(intermed$BM3_NEW >= 30,
                                  1,
                                  NA))

outcomes$AW3_NEW <- ifelse(intermed$AW1 == 0 |
                             (intermed$AW1 == 1 &
                                intermed$AW2_COD == 0),
                           0,
                           ifelse(intermed$AW1 == 1 &
                                    intermed$AW2_COD == 1,
                                  1,
                                  NA))
outcomes$PL8_NEW <- ifelse((intermed$PL6 == 0 &
                              intermed$IN9 == 0 &
                              intermed$PL7 == 0) |
                             ((intermed$PL6 == 1 |
                                 intermed$IN9 == 1 |
                                 intermed$PL7 == 1 ) &
                                intermed$IN14==0),
                           0,
                           ifelse((intermed$PL6 == 1 |
                                     intermed$IN9 == 1 |
                                     intermed$PL7 == 1) |
                                    intermed$IN14 == 1,
                                  1,
                                  NA))
outcomes$PC1_NEW <- ifelse(apply(intermed[, c("PC1A", "PC1B", "PC1C", "PC1D", "PC1E",
                                              "IN15")],
                                 1,
                                 sum)==0,
                           0,
                           ifelse(apply(intermed[, c("PC1A", "PC1B", "PC1C", "PC1D", "PC1E",
                                                     "IN15")],
                                        1,
                                        sum,
                                        na.rm = TRUE) >= 1,
                                  1,
                                  NA))
outcomes$HA1_NEW <- ifelse(apply(intermed[, c("HA1A", "HA1B", "HA1C", "HA1D", "HA1E", "HA1F",
                                              "HA1G")],
                                 1,
                                 sum) == 0,
                           0,
```

```
                          ifelse(apply(intermed[, c("HA1A", "HA1B", "HA1C", "HA1D", "HA1E",
                                                    "HA1F", "HA1G")],
                                       1,
                                       sum,
                                       na.rm = TRUE) >= 1,
                                 1,
                                 NA))
outcomes$IN61_NEW <- ifelse(intermed$IN60_NEW <= median(intermed$IN60_NEW,
                                                        na.rm=TRUE),
                            0,
                            ifelse(intermed$IN60_NEW > median(intermed$IN60_NEW,
                                                              na.rm=TRUE),
                                   1,
                                   NA))

others$HEALTHY_NEW <- ifelse(apply(diseases[, -1],
                                   1,
                                   sum) >= 1,
                             0,
                             ifelse(apply(diseases[, -1],
                                          1,
                                          sum) == 0,
                                    1,
                                    NA))
others$ILL_NEW <- ifelse(apply(diseases[, -1],
                               1,
                               sum) >= 1,
                         1,
                         ifelse(apply(diseases[, -1],
                                      1,
                                      sum) == 0,
                                0,
                                NA))
others$JOB_NEW <- ifelse(is.na(intermed$MAINSTAT),
                         NA,
                         ifelse(intermed$MAINSTAT == 10,
                                1,
                                0))

others$OVER65_NEW <- ifelse(conf$AGE_COD >= 11,
                            1,
                            ifelse(conf$AGE_COD < 11,
                                   0,
                                   NA))


# Reordering all variables for the final data frames

diseases <- diseases[, c("PID", "CD1A", "CD1B", "CD1C", "CD1D", "CD1E", "CD1F", "CD1G",
                         "CD1HI_NEW", "CD1J", "CD1K", "CD1L", "CD1M", "CD1N", "CD1O",
                         "PL2_COD", "PL6_NEW", "BM4_NEW")]

outcomes <- outcomes[, c("PID", "HS1", "AW3_NEW", "PL8_NEW", "PC1_NEW", "HA1_NEW", "PN1",
                         "PN2", "HO1", "HO3", "AM3", "AM5", "MD1", "MD2", "IN61_NEW")]

conf <- conf[, c("PID", "SEX", "AGE_COD", "HATLEVEL_COD", "HHINCOME")]
```

```r
others <- others[, c("PID", "WGT", "REGION", "HEALTHY_NEW", "ILL_NEW", "JOB_NEW",
                     "OVER65_NEW")]


# Selection of individuals with complete data for all diseases

index <- complete.cases(diseases)
diseases <- diseases[index, ]
outcomes <- outcomes[index, ]
conf <- conf[index, ]
others <- others[index, ]


# Individuals excluded due to missing data for any disease

MOT <- rep("diseases",
           sum(!index))
excl <- cbind(PID = diseases[!index, 1],
              MOT,
              diseases[!index, -1],
              outcomes[!index, -1],
              conf[!index, -1],
              others[!index, -1])


# Description of variables

diseases_labels <- data.frame(Code = colnames(diseases),
                              Label = c("Identification number",
                                        "Asthma",
                                        "Chronic pulmonary disease",
                                        "Myocardial infarction",
                                        "Coronary heart disease",
                                        "Hypertension",
                                        "Stroke",
                                        "Arthrosis",
                                        "Low back/neck disorder",
                                        "Diabetes",
                                        "Allergy",
                                        "Cirrhosis",
                                        "Urinary incontinence",
                                        "Kidney problems",
                                        "Depression",
                                        "Visual impairment",
                                        "Hearing impairment",
                                        "Obesity"),
                              stringsAsFactors = FALSE)

outcomes_labels <- data.frame(Code = colnames(outcomes),
                              Label = c("Identification number",
                                        "Negative self-perceived general health",
                                        "Absence from work (only employed individuals)",
                                        "Physical functional limitations",
                                        "Personal care activities limitations
                                         (\u003E65 years old)",
```

```
                                        "Household activities limitations
                                        (\u003E65 years old)",
                                        "Intensity of bodily pain",
                                        "Impact of the pain on daily life",
                                        "Hospitalisation as inpatient",
                                        "Hospitalisation as day patient",
                                        "Consultation of a general practitioner",
                                        "Consultation of other specialist",
                                        "Use of medicines prescribed",
                                        "Use of medicines not prescribed",
                                        "Out-of-pocket health expenditure"),
                                stringsAsFactors = FALSE)

conf_labels <- data.frame(Code = colnames(conf),
                          Label = c("Identification number",
                                    "Sex",
                                    "Age",
                                    "Level of education",
                                    "Net monthly income"),
                          stringsAsFactors = FALSE)

others_labels <- data.frame(Code=colnames(others),
                            Label=c("Identification number",
                                    "Weight",
                                    "NUTS II (2002) Region",
                                    "Healthy subjects",
                                    "Any chronic disease",
                                    "Employed individuals",
                                    "Over 65 years old"),
                            stringsAsFactors = FALSE)

excl_labels <- rbind(diseases_labels[1, ],
                     c("MOT",
                       "Exclusion criteria"),
                     diseases_labels[-1, ],
                     outcomes_labels[-1, ],
                     conf_labels[-1, ],
                     others_labels[-1, ])


# Creation of final lists

data <- list(diseases = diseases,
             outcomes = outcomes,
             conf = conf,
             others = others,
             excl = excl)

labels <- list(diseases_labels = diseases_labels,
               outcomes_labels = outcomes_labels,
               conf_labels = conf_labels,
               others_labels = others_labels,
               excl_labels = excl_labels)

ins2014 <- list(data = data,
                labels = labels)
```

```
  # Save the new databases to .csv files

  write.csv(diseases,
            "diseases.csv",
            row.names = FALSE)
  write.csv(outcomes,
            "outcomes.csv",
            row.names = FALSE)
  write.csv(conf,
            "conf.csv",
            row.names = FALSE)
  write.csv(others,
            "others.csv",
            row.names = FALSE)
  write.csv(excl,
            "excl.csv",
            row.names = FALSE)
  write.csv(diseases_labels,
            "diseases_labels.csv",
            row.names = FALSE)
  write.csv(outcomes_labels,
            "outcomes_labels.csv",
            row.names = FALSE)
  write.csv(conf_labels,
            "conf_labels.csv",
            row.names = FALSE)
  write.csv(others_labels,
            "others_labels.csv",
            row.names = FALSE)
  write.csv(excl_labels,
            "excl_labels.csv",
            row.names = FALSE)

  rm(intermed, diseases, outcomes, conf, others, index, MOT, excl, diseases_labels,
     outcomes_labels, conf_labels, others_labels, excl_labels, data, labels)
  gc()


  # End the function and return the list with data and data descriptions

  return(ins2014)
}
```

To perform the data extraction and recoding, and compute the previous function, the following code was used:

```
ins2014 <- fun01_readandrecodefun(datafile = "INS2014_anonimizada_20160308.sav",
                                  workingdirectory = "C:\\INS2014")
```

Exploratory analysis

For the exploratory analysis, three functions were created: `fun02_pcamodelfun`, `fun03_pcasummaryfun` and `fun04_pcaclassesfun`.

The `fun02_pcamodelfun` function computes the principal component analysis (PCA). This function has as dependency the `psych` package.

It has as arguments:

Arguments	Default	Description
`x`	ins2014	Output object from function `fun01_readandrecodefun`
`rot`	"varimax"	Type of rotation to be used in PCA (options: "none", "varimax", "quatimax", "promax", "oblimin", "simplimax" and "cluster")
`cutoff`	0.4	Threshold to consider in the analysis of factor loadings

It has as output:

- list with PCA models for all possible component numbers (from one to the total number of diseases).

```
fun02_pcamodelfun <- function(x = ins2014,
                              rot = "varimax",
                              cutoff = 0.4) {


  #### Function for computing the Principal Component Analysis (PCA) with a correlation
  #### matrix using a Pearson correlation coefficient, equivalent to the phi coefficient
  #### for dichotomous variables, and "varimax" rotation (or other, or "none")

  # If necessary install the psych package and load it

  packages <- c("psych")
  newpackages <- packages[!(packages %in% installed.packages()[, "Package"])]
  if(length(newpackages)) install.packages(newpackages)
  library(psych)


  # PCA with the defined rotation and threshold

  pcamodel <- vector("list",
                     ncol(x$data$diseases[, -1]))

  for (i in 1:ncol(x$data$diseases[, -1])) {
    pcamodel[[i]] <- principal(r = x$data$diseases[, -1],
                               nfactors = i,
                               rotate = rot,
                               covar = FALSE)
  }

  for (i in 1:length(pcamodel)) {
    pcamodel[[i]]$components <- ifelse(pcamodel[[i]]$loadings >= cutoff,
                                       "+",
                                       ifelse(pcamodel[[i]]$loadings <= -cutoff,
                                              "-",
                                              " "))
    rownames(pcamodel[[i]]$components) <- x$labels$diseases_labels[-1, 2]
    p <- print(pcamodel[[i]])
```

```
    pcamodel[[i]]$summary <- p$Vaccounted
  }

  rm(packages, newpackages, p)
  gc()


  # End the function and return the PCA model

  return(pcamodel)
}
```

The `fun03_pcasummaryfun` function computes some outputs for deciding the number of components to use in the PCA. This function has no additional dependencies besides base R packages.

It has as arguments:

Arguments	Default	Description
`model`	pcamodel	Output object from function `fun02_pcamodelfun`
`ncomp`	8	Number of principal components to consider in the outputs

It has as outputs:

- Scree plot of PCA;
- table with factor loadings of each disease, by component;
- table with the correspondence of the diseases to each component, according to the considered ~threshold in the function `fun02_pcamodelfun` (the assignment is marked with the symbols `+` if positive correlation or `-` if negative correlation with the component).

```
fun03_pcasummaryfun <- function(model = pcamodel,
                                ncomp = 8) {


  #### Function to produce summary outputs of the PCA to decide the number of components
  #### (scree plot and tables with loading factors and distribution of diseases by component)


  # Scree plot

  plot(model[[1]]$values,
       main = "Scree plot",
       xlab = "Number of components",
       ylab = "Eigenvalue",
       xaxt = 'n',
       pch = 16,
       cex = 0.6,
       cex.axis = 0.7)

  lines(model[[1]]$values,
        lty = 1)

  axis(side = 1,
       at = 1:length(model[[1]]$values),
       labels = 1:length(model[[1]]$values),
```

```
      cex.axis = 0.7)

  abline(1,
         0,
         lty = 3)


  # Table with factor loadings

  print(model[[ncomp]]$summary)


  # Table with the distribution of diseases by component

  print(model[[ncomp]]$components)
}
```

The `fun04_pcaclassesfun` function computes the creation of possible classes of sufficient causes based on the number of selected components (all possible combinations of diseases by component) and distributes the individuals by the created components and classes. This function has no additional dependencies besides base R packages.

It has as arguments:

Arguments	Default	Description
`x`	ins2014	Output object from function `fun01_readandrecodefun`
`model`	pcamodel	Output object from function `fun02_pcamodelfun`
`ncomp`	8	Number of principal components to consider in the outputs
`workingdirectory`	getwd()	Name of the folder previously passed to the `datafile` argument of the function `fun01_readandrecodefun` (between quotation marks and with the \\ or / symbol separating each level, eg: "C:\INS2014")

It has as outputs:

- *list* containing two *lists*, `data` and `labels`, each containing seven *data frames*:
  - `data$classes`: distribution of individuals by the created possible classes of sufficient causes,
  - `data$classes_model`: distribution of individuals by the created possible classes of sufficient causes, organized so that they can be used in the functions of the confirmatory analysis,
  - `data$components`: distribution of individuals by the created components,
  - `data$diseases`: independent variables (diseases),
  - `data$outcomes`: dependent variables (morbidity indicators),
  - `data$conf`: confounding variables,
  - `data$others`: other variables of interest for the analysis (such as weights and regions),
  - `data$excl`: individuals excluded from the analysis (due to missing data on one or more diseases),
  - `labels$classes_labels`: description of variables in `data$classes`,
  - `labels$classes_model_labels`: description of variables in `data$classes_model`,
  - `labels$components_labels`: description of variables in `data$components`,
  - `labels$diseases_labels`: description of variables in `data$diseases`,
  - `labels$outcomes_labels`: description of variables in `data$outcomes`,

  - `labels$conf_labels`: description of variables in `data$conf`,
  - `labels$others_labels`: description of variables in `data$others`,
  - `labels$excl_labels`: description of variables in `data$excl`;
- six `.csv` files with the information contained in each of the previous *data frames* (`data$classes`, `data$classes_model`, `data$components`, `labels$classes_labels`, `labels$classes_model_labels` and `labels$components_labels`), saved in the folder passed to the `workingdirectory` argument.

```
fun04_pcaclassesfun <- function (x = ins2014,
                                 model = pcamodel,
                                 ncomp = 8,
                                 workingdirectory = getwd()) {


  #### Creation of possible classes of sufficient causes based on the number of components


  # Open the appropriate working directory

  setwd(workingdirectory)


  # Creating an index for classes

  indexpca <- apply(model[[ncomp]]$components,
                    2,
                    function(y) which(y == "+"))

  lengthindexpca <- lapply(indexpca,
                           length)


  # Creation of all possible combinations of diseases by component (classes)

  indexclasses <- list()


  indexcompl <- list()

  classes_model_index <- list()

  for (i in 1:length(indexpca)) {
    y <- list()
    a <- list()
    lengthindexcompl <- length(indexcompl)
    if (length(indexpca[[i]]) == 1) {
      y[[1]] <- indexpca[[i]]
      a[[1]] <- NA
    } else {
      z <- list()
      b <- list()
      temp <- NULL
      for (j in length(indexpca[[i]]):1) {
        z <- combn(indexpca[[i]],
                   j,
                   simplify = FALSE)
        temp <- c(temp, length(z))
```

```
      if (j == length(indexpca[[i]])) {
        b[[1]] <- NA
      } else {
        b <- rep(list(1:sum(temp[-length(temp)]) + lengthindexcompl),
                 length(z))
      }
      y <- c(y,
             z)
      a <- c(a,
             b)
    }
  }
  classes_model_index[[i]] <- rep(i,
                                  times = length(y))
  indexclasses <- c(indexclasses,
                    y)
  indexcompl <- c(indexcompl,
                  a)
}


# Distribution of individuals by classes and components

classes <- x$data$diseases$PID

for (i in 1:length(indexclasses)) {
  if (length(indexclasses[[i]]) == 1) {
    y <-  x$data$diseases[, -1][, indexclasses[[i]]]
  } else {
    y <- apply(x$data$diseases[, -1][, indexclasses[[i]]],
               1,
               prod)
  }
  classes <- cbind(classes,
                   y)
}

for (i in 1:ncol(classes[, -1])) {
  if (is.na(indexcompl[[i]][1])) {
    next
  } else {
    for (j in 1:length(indexcompl[[i]])) {
      classes[, -1][, i] <- ifelse(classes[, -1][, indexcompl[[i]][j]] == 1,
                                   0,
                                   classes[, -1][, i])
    }
  }
}

components <- data.frame(PID = x$data$diseases$PID)

for (i in 1:length(indexpca)) {
  components[, i + 1] <- ifelse(rowSums(as.data.frame(
    x$data$diseases[, indexpca[[i]]+1])
  ) > 0,
  1,
```

```
  0)
}


# Adjusting class and component names

colnames(classes) <- c(as.character(x$labels$diseases_labels[1, 1]),
                       unlist(lapply(indexclasses,
                                     function(y) {
                                       paste(x$labels$diseases_labels[-1, 1][y],
                                             collapse = "-")
                                     }
                       )))

classes_labels <- data.frame(Code = colnames(classes),
                             Label = c(as.character(x$labels$diseases_labels[1, 2]),
                                       unlist(lapply(indexclasses,
                                                     function(y) {
                                                       paste(x$labels$diseases_labels
                                                             [-1, 2][y],
                                                             collapse = ", ")
                                                     }
                                       ))))

classes_labels <- as.data.frame(apply(classes_labels,
                                      2,
                                      as.character),
                                stringsAsFactors = FALSE)

comma <- sapply(classes_labels[, 2],
                function(x) unlist(gregexpr(x,
                                            pattern = ",")))

comma <- lapply(comma,
                function(x) x[length(x)])

for (i in 1:length(comma)){
  if (comma[[i]] > 0) {
    classes_labels[i, 2] <- paste(substr(classes_labels[i, 2],
                                         1,
                                         1),
                                  tolower(substr(classes_labels[i, 2],
                                                 2,
                                                 comma[[i]] - 1)),
                                  " e",
                                  tolower(substr(classes_labels[i, 2],
                                                 comma[[i]] + 1,
                                                 nchar(classes_labels[i, 2]))),
                                  sep = "")
  } else next
}

colnames(components) <- c(as.character(x$labels$diseases_labels[1, 1]),
                          unlist(lapply(indexpca,
                                        function(y) {
                                          paste(x$labels$diseases_labels[-1, 1][y],
```

```
                                            collapse = "-")
                                    }
                            )))

components_labels <- data.frame(Code = colnames(components),
                                Label = c(as.character(x$labels$diseases_labels[1, 2]),
                                          unlist(lapply(indexpca,
                                                        function(y) {
                                                          paste(x$labels$diseases_labels
                                                                [-1, 2][y],
                                                                collapse = ", ")
                                                        }
                                          ))))

components_labels <- as.data.frame(apply(components_labels,
                                         2,
                                         as.character),
                                   stringsAsFactors = FALSE)

comma <- sapply(components_labels[, 2],
                function(x) unlist(gregexpr(x,
                                            pattern = ",")))

comma <- lapply(comma,
                function(x) x[length(x)])

for (i in 1:length(comma)){
  if (comma[[i]] > 0) {
    components_labels[i, 2] <- paste(substr(components_labels[i, 2],
                                            1,
                                            1),
                                     tolower(substr(components_labels[i, 2],
                                                    2,
                                                    comma[[i]] - 1)),
                                     " e/ou",
                                     tolower(substr(components_labels[i, 2],
                                                    comma[[i]] + 1,
                                                    nchar(components_labels[i, 2]))),
                                     sep = "")
  } else next
}

classes_labels$component <- c(NA,
                              unlist(classes_model_index))

classes_labels$component_label <- c(NA,
                                    components_labels[-1, ][unlist(classes_model_index), 2])


# Adjust the reference as not belonging to the PCA component

classes_model <- classes

classes_model_index <- unlist(classes_model_index)

for (i in 2:ncol(classes_model)) {
```

```
  classes_model[, i] <- ifelse(classes_model[, i] == 1,
                               1,
                               ifelse(components[, classes_model_index[i - 1] + 1] == 0,
                                      0,
                                      NA))
}


# Creation of the final list

data <- list(classes = classes,
             classes_model = classes_model,
             components = components,
             diseases = x$data$diseases,
             outcomes = x$data$outcomes,
             conf = x$data$conf,
             others = x$data$others,
             excl = x$data$excl)

labels <- list(classes_labels = classes_labels,
               classes_model_labels = classes_labels,
               components_labels = components_labels,
               diseases_labels = x$labels$diseases_labels,
               outcomes_labels = x$labels$outcomes_labels,
               conf_labels = x$labels$conf_labels,
               others_labels = x$labels$others_labels,
               excl_labels = x$labels$excl_labels)

pcaclasses <- list(data = data,
                   labels = labels)



# Save the new data to .csv files

write.csv(classes,
          "classes.csv",
          row.names = FALSE)
write.csv(classes_model,
          "classes_model.csv",
          row.names = FALSE)
write.csv(components,
          "components.csv",
          row.names = FALSE)
write.csv(classes_labels,
          "classes_labels.csv",
          row.names = FALSE)
write.csv(classes_labels,
          "classes_model_labels.csv",
          row.names = FALSE)
write.csv(components_labels,
          "components_labels.csv",
          row.names = FALSE)

rm(indexpca, lengthindexpca, indexclasses, indexcompl, classes_model_index, y, a,
   lengthindexcompl, z, b, temp, classes, components, classes_labels, comma,
```

```
    components_labels, classes_model, data, labels)
  gc()


  # End the function and return the final list

  return(pcaclasses)
}
```

To perform the exploratory analysis and compute the three previous functions, the following code was used:

```
pcamodel <- fun02_pcamodelfun(x = ins2014,
                              rot = "varimax",
                              cutoff = 0.4)

fun03_pcasummaryfun(model = pcamodel,
                    ncomp = 8)

pcaclasses <- fun04_pcaclassesfun(x = ins2014,
                                  model = pcamodel,
                                  ncomp = 8,
                                  workingdirectory = "C:\\INS2014")
```

Confirmatory analysis

For the confirmatory analysis, three functions were created: `fun05_pcapaffun`, `fun06_pafregionfun` and `fun07_paflevelsfun`.

The `fun05_pcapaffun` function computes the confirmatory statistical analysis of the study. This function has as dependency the `questionr` package.

It has as arguments:

Arguments	Default	Description
`x`	pcaclasses	Output object from function `fun04_pcaclassesfun`
`level`	Sem pré-definição	Level of analysis to be performed (whether by components or by classes)
`weights`	`TRUE`	Logical argument (`TRUE` or `FALSE`), for the use (or not) of the weights in the analysis (respectively)
`adjustment`	`TRUE`	Logical argument (`TRUE` or `FALSE`), to adjust (or not) to other components in the analysis (respectively)

It has as outputs:

- *list* containing two *data frames*:
  - `results`: computed statistics,
  - `results_labels`: description of the statistics listed in `results`.

```
fun05_pcapaffun <- function(x = pcaclasses,
                            level = c("components",
```

```
                                "classes"),
                        weights = TRUE,
                        adjustment = TRUE) {


#### Confirmatory statistical analysis taking into account the components
#### and possible classes of sufficient causes, according to the PCA


# Level of analysis

if (level == "components") {
  leveldata <- x$data$components
  leveldata_model <- leveldata
} else if (level == "classes") {
  leveldata <- x$data$classes
  leveldata_model <- x$data$classes_model
} else {
  stop("Invalid 'adjustment' argument (should be 'components' or 'classes')")
}


# Verification of arguments

if (adjustment != TRUE & adjustment != FALSE) {
  stop("Invalid 'adjustment' argument (should be TRUE or FALSE)")
}
if (weights == TRUE) {
  wgt  <- x$data$others$WGT
} else if (weights == FALSE) {
  wgt <- NULL
} else {
  stop("Invalid 'weights' argument (should be TRUE or FALSE)")
}


# If necessary install the questionr package and load it

packages <- c("questionr")
newpackages <- packages[!(packages %in% installed.packages()[, "Package"])]
if(length(newpackages)) install.packages(newpackages)
library(questionr)


# Look for complete cases in the confounding and outcome variables

nclasses <- ncol(leveldata[, -1])
noutcomes <- ncol(x$data$outcomes[, -1])
confcomplcases <- which(complete.cases(x$data$conf))
outcomescomplcases <- apply(x$data$outcomes,
                            2,
                            function(x) !is.na(x))
indexcomplcases <- apply(outcomescomplcases,
                         2,
                         function(x) as.logical(x*confcomplcases))
```

```
# Calculation of some statistics

n_total_classes_temp <- apply(leveldata[, -1],
                              2,
                              length)
n_total_classes <- rep(c(length(x$data$others$ILL_NEW),
                         n_total_classes_temp),
                       times = noutcomes)
n_classes_temp <- apply(leveldata[, -1],
                        2,
                        sum)
n_classes <- rep(c(sum(x$data$others$ILL_NEW),
                   n_classes_temp),
                 times = noutcomes)
n_total_outcomes_temp <- apply(indexcomplcases[, -1],
                               2,
                               sum)
n_total_outcomes <- rep(n_total_outcomes_temp,
                        each = nclasses + 1)
n_outcomes_temp <- apply(x$data$outcomes[, -1],
                         2,
                         sum,
                         na.rm = TRUE)
n_outcomes <- rep(n_outcomes_temp,
                  each = nclasses + 1)
tab_wgt_classes <- apply(cbind(x$data$others$ILL_NEW,
                               leveldata[, -1]),
                         2,
                         wtd.table,
                         weights = wgt)
prop_classes_pop_temp <- apply(tab_wgt_classes,
                               2,
                               prop.table)["1", ]
prop_classes_pop <- rep(prop_classes_pop_temp,
                        times = noutcomes)


# Index of components for classes and components

if (level == "components") {
  comp_index <- rep(0:nclasses,
                    times = noutcomes)
} else if (level == "classes") {
  comp_index <- rep(c(0,
                      x$labels$classes_labels[-1, 3]),
                    times = noutcomes)
}


# Calculation of log poisson model and other statistics

model <- list()
if (adjustment == TRUE) {model_adj <- list()}

n_classes_outcomes <- list()
```

```
prop_outcomes_pop <- list()
prop_outcomes_classes <- list()
prop_classes_outcomes <- list()

for (i in 2:(noutcomes + 1)) {
  model_temp <- list(
    glm(formula = x$data$outcomes[indexcomplcases[, i], i] ~
          x$data$others$ILL_NEW[indexcomplcases[, i]] +
          .,
        data = as.data.frame(x$data$conf[indexcomplcases[, i], -1]),
        family = poisson(link = log),
        weight = wgt[indexcomplcases[, i]],
        na.action = na.exclude,
        control = list(maxit = 100))
  )

  if (adjustment == TRUE) {model_adj_temp <- model_temp}

  tab_out_cla_ill <- table(x$data$outcomes[indexcomplcases[, i], i],
                           x$data$others$ILL_NEW[indexcomplcases[, i]])
  tab_out_cla_wgt_ill <- wtd.table(x$data$outcomes[indexcomplcases[, i], i],
                                   x$data$others$ILL_NEW[indexcomplcases[, i]],
                                   weights = wgt[indexcomplcases[, i]])

  try_tab_out_cla_1 <- class(try(tab_out_cla_ill["1", ],
                                 silent = TRUE)) == "try-error"
  try_tab_out_cla_2 <- class(try(tab_out_cla_ill[, "1"],
                                 silent = TRUE)) == "try-error"
  try_tab_out_cla_1_2 <- try_tab_out_cla_1 & try_tab_out_cla_2

  if (try_tab_out_cla_1_2) {
    n_classes_outcomes_temp <- list(0)
    prop_outcomes_pop_temp <- list(0)
    prop_outcomes_classes_temp <- list(NA)
    prop_classes_outcomes_temp <- list(NA)
  } else if (try_tab_out_cla_1) {
    n_classes_outcomes_temp <- list(0)
    prop_outcomes_pop_temp <- list(0)
    prop_outcomes_classes_temp <- list(0)
    prop_classes_outcomes_temp <- list(NA)
  } else if (try_tab_out_cla_2) {
    n_classes_outcomes_temp <- list(0)
    prop_outcomes_pop_temp <- list(sum(prop.table(tab_out_cla_wgt_ill)["1", ]))
    prop_outcomes_classes_temp <- list(NA)
    prop_classes_outcomes_temp <- list(0)
  } else {
    n_classes_outcomes_temp <- list(tab_out_cla_ill["1", "1"])
    prop_outcomes_pop_temp <- list(sum(prop.table(tab_out_cla_wgt_ill)["1", ]))
    prop_outcomes_classes_temp <- list(prop.table(tab_out_cla_wgt_ill,
                                                  margin = 2)["1", "1"])
    prop_classes_outcomes_temp <- list(prop.table(tab_out_cla_wgt_ill,
                                                  margin = 1)["1", "1"])
  }

  for (j in 2:(nclasses + 1)) {
    model_temp[[j]] <- glm(formula = x$data$outcomes[indexcomplcases[, i], i] ~
```

```
                          leveldata_model[indexcomplcases[, i], j] +
                          .,
                        data = as.data.frame(x$data$conf[indexcomplcases[, i], -1]),
                        family = poisson(link = log),
                        weight = wgt[indexcomplcases[, i]],
                        na.action = na.exclude,
                        control = list(maxit = 100))

  if (adjustment == TRUE) {
    data <- cbind(x$data$components[indexcomplcases[, i], -1],
                  x$data$conf[indexcomplcases[, i], -1])
    if (level == "components") {
      model_adj_temp[[j]] <- glm(formula = x$data$outcomes[indexcomplcases[, i], i] ~
                                   leveldata_model[indexcomplcases[, i], j] +
                                   .,
                                 data = as.data.frame(data)[, -(j-1)],
                                 family = poisson(link = log),
                                 weight = wgt[indexcomplcases[, i]],
                                 na.action = na.exclude,
                                 control = list(maxit = 100))
    } else if (level == "classes") {
      model_adj_temp[[j]] <- glm(formula = x$data$outcomes[indexcomplcases[, i], i] ~
                                   leveldata_model[indexcomplcases[, i], j] +
                                   .,
                                 data = as.data.frame(data)
                                 [, -x$labels$classes_labels[j, 3]],
                                 family = poisson(link = log),
                                 weight = wgt[indexcomplcases[, i]],
                                 na.action = na.exclude,
                                 control = list(maxit = 100))
    }
  }

  tab_out_cla <- table(x$data$outcomes[indexcomplcases[, i], i],
                       leveldata[indexcomplcases[, i], j])
  tab_out_cla_wgt <- wtd.table(x$data$outcomes[indexcomplcases[, i], i],
                               leveldata[indexcomplcases[, i], j],
                               weights = wgt[indexcomplcases[, i]])

  try_tab_out_cla_1 <- class(try(tab_out_cla["1", ],
                                 silent = TRUE)) == "try-error"
  try_tab_out_cla_2 <- class(try(tab_out_cla[, "1"],
                                 silent = TRUE)) == "try-error"
  try_tab_out_cla_1_2 <- try_tab_out_cla_1 & try_tab_out_cla_2

  if (try_tab_out_cla_1_2) {
    n_classes_outcomes_temp[[j]] <- 0
    prop_outcomes_pop_temp[[j]] <- 0
    prop_outcomes_classes_temp[[j]] <- NA
    prop_classes_outcomes_temp[[j]] <- NA
  } else if (try_tab_out_cla_1) {
    n_classes_outcomes_temp[[j]] <- 0
    prop_outcomes_pop_temp[[j]] <- 0
    prop_outcomes_classes_temp[[j]] <- 0
    prop_classes_outcomes_temp[[j]] <- NA
  } else if (try_tab_out_cla_2) {
```

```
      n_classes_outcomes_temp[[j]] <- 0
      prop_outcomes_pop_temp[[j]] <- sum(prop.table(tab_out_cla_wgt)["1", ])
      prop_outcomes_classes_temp[[j]] <- NA
      prop_classes_outcomes_temp[[j]] <- 0
    } else {
      n_classes_outcomes_temp[[j]] <- tab_out_cla["1", "1"]
      prop_outcomes_pop_temp[[j]] <- sum(prop.table(tab_out_cla_wgt)["1", ])
      prop_outcomes_classes_temp[[j]] <- prop.table(tab_out_cla_wgt,
                                                    margin = 2)["1", "1"]
      prop_classes_outcomes_temp[[j]] <- prop.table(tab_out_cla_wgt,
                                                    margin = 1)["1", "1"]
    }

    rm(data, tab_out_cla,
       tab_out_cla_wgt,
       try_tab_out_cla_1,
       try_tab_out_cla_2,
       try_tab_out_cla_1_2)
    gc()
  }

  model[[i-1]] <- model_temp
  if (adjustment == TRUE) {model_adj[[i-1]] <- model_adj_temp}
  n_classes_outcomes[[i-1]] <- n_classes_outcomes_temp
  prop_outcomes_pop[[i-1]] <- prop_outcomes_pop_temp
  prop_outcomes_classes[[i-1]] <- prop_outcomes_classes_temp
  prop_classes_outcomes[[i-1]] <- prop_classes_outcomes_temp

  rm(model_temp,
     tab_out_cla_ill,
     tab_out_cla_wgt_ill,
     n_classes_outcomes_temp,
     prop_outcomes_pop_temp,
     prop_outcomes_classes_temp,
     prop_classes_outcomes_temp)
  if (adjustment == TRUE) {rm(model_adj_temp)}
  gc()
}


# Calculation of prevalence ratios (PR)

pr <- unlist(lapply(model, function(y) {
  lapply(y, function(z) {
    if (class(try(coef(summary(z))[2, 1],
                  silent = TRUE)) == "try-error") {
      NA
    } else {
      exp(coef(summary(z))[2, 1])
    }
  })
}))


# Extraction of p values
```

```r
pvalue <- unlist(lapply(model, function(y) {
  lapply(y, function(z) {
    if (class(try(coef(summary(z))[2, 4],
                  silent = TRUE)) == "try-error") {
      NA
    } else {
      coef(summary(z))[2, 4]
    }
  })
}))


# Adjustment of results

n_classes_outcomes <- unlist(n_classes_outcomes)
prop_classes_pop <- round(prop_classes_pop * 100,
                          2)
prop_outcomes_pop <- round(unlist(prop_outcomes_pop) * 100,
                           2)
prop_outcomes_classes <- round(unlist(prop_outcomes_classes) * 100,
                               2)
prop_classes_outcomes <- round(unlist(prop_classes_outcomes) * 100,
                               2)
pr <- ifelse(is.na(pr),
             NA,
             round(pr,
                   2))
pvalue <- ifelse(is.na(pvalue),
                 NA,
                 ifelse(pvalue < 0.001,
                        "<0.001",
                        round(pvalue,
                              3)))


# Calculation of population attributable fractions (PAF)

paf <- ifelse(!is.na(prop_classes_outcomes) & !is.na(pr),
              round(prop_classes_outcomes / 100 * (pr - 1) / (pr) * 100,
                    2),
              NA)


# Organization of results and labels

if (level == "components") {
  results <- data.frame(outcomes = rep(x$labels$outcomes_labels[-1, 2],
                                       each = nclasses + 1),
                        components = rep(c("Any chronic disease",
                                           x$labels$components_labels[-1, 2]),
                                         times = noutcomes),
                        n_total_components = n_total_classes,
                        n_components = n_classes,
                        n_total_outcomes = n_total_outcomes,
                        n_outcomes = n_outcomes,
                        n_components_outcomes = n_classes_outcomes,
```

```
                        prop_components_pop = prop_classes_pop,
                        prop_outcomes_pop = prop_outcomes_pop,
                        prop_outcomes_components = prop_outcomes_classes,
                        prop_components_outcomes = prop_classes_outcomes,
                        pr = pr,
                        pvalue = pvalue,
                        paf = paf)

  results_labels <- data.frame(Code = colnames(results),
                        Label = c("Morbidity indicator",
                                  "Disease(s)",
                                  "Total number of subjects (components)",
                                  "Number of subjects with the disease(s)",
                                  "Total number of subjects (outcome)",
                                  "Number of subjects with the outcome",
                                  "Number of subjects with the disease(s) and the outcome",
                                  "Proportion (%) of subjects with the disease(s)
                                  in the population",
                                  "Proportion (%) of subjects with the outcome
                                  in the population",
                                  "Proportion (%) of subjects with the outcome
                                  within those with the disease(s)",
                                  "Proportion (%) of subjects with the disease(s)
                                  within those with the outcome",
                                  "Prevalence ratio",
                                  "p value",
                                  "Population Attributable Fraction (%)"))

} else if (level == "classes") {
  results <- data.frame(outcomes = rep(x$labels$outcomes_labels[-1, 2],
                                       each = nclasses + 1),
                        classes = rep(c("Any chronic disease",
                                        x$labels$classes_labels[-1, 2]),
                                      times = noutcomes),
                        n_total_classes = n_total_classes,
                        n_classes = n_classes,
                        n_total_outcomes = n_total_outcomes,
                        n_outcomes = n_outcomes,
                        n_classes_outcomes = n_classes_outcomes,
                        prop_classes_pop = prop_classes_pop,
                        prop_outcomes_pop = prop_outcomes_pop,
                        prop_outcomes_classes = prop_outcomes_classes,
                        prop_classes_outcomes = prop_classes_outcomes,
                        pr = pr,
                        pvalue = pvalue,
                        paf = paf)

  results_labels <- data.frame(Code = colnames(results),
                        Label = c("Morbidity indicator",
                                  "Disease(s)",
                                  "Total number of subjects (components)",
                                  "Number of subjects with the disease(s)",
                                  "Total number of subjects (outcome)",
                                  "Number of subjects with the outcome",
                                  "Number of subjects with the disease(s) and the outcome",
                                  "Proportion (%) of subjects with the disease(s)
```

```
                                  in the population",
                                  "Proportion (%) of subjects with the outcome
                                  in the population",
                                  "Proportion (%) of subjects with the outcome
                                  within those with the disease(s)",
                                  "Proportion (%) of subjects with the disease(s)
                                  within those with the outcome",
                                  "Prevalence ratio",
                                  "p value",
                                  "Population Attributable Fraction (%)"))
}


# Calculation of results for adjustment = TRUE

if (adjustment == TRUE) {

  # Calculation of adjusted prevalence ratios (PR)
  pr_adj <- unlist(lapply(model_adj,
                          function(y) {
                            lapply(y, function(z) {
                              if (class(try(coef(summary(z))[2, 1],
                                            silent = TRUE)) == "try-error") {
                                NA
                              } else {
                                exp(coef(summary(z))[2, 1])
                              }
                            })
                          }))

  # Extraction of the adjusted p values
  pvalue_adj <- unlist(lapply(model_adj,
                              function(y) {
                                lapply(y,
                                       function(z) {
                                         if (class(try(coef(summary(z))[2, 4],
                                                       silent = TRUE)) == "try-error") {
                                           NA
                                         } else {
                                           coef(summary(z))[2, 4]
                                         }
                                       })
                              }))

  # Adjustment of results
  results$pr_adj <- ifelse(is.na(pr_adj),
                           NA,
                           round(pr_adj,
                                 2))
  results$pvalue_adj <- ifelse(is.na(pvalue_adj),
                               NA,
                               ifelse(pvalue_adj < 0.001,
                                      "<0.001",
                                      round(pvalue_adj,
                                            3)))
  results$paf_adj <- ifelse(!is.na(prop_classes_outcomes) & !is.na(results$pr_adj),
```

```
                              round(prop_classes_outcomes / 100 * (results$pr_adj - 1) /
                                      (results$pr_adj) * 100,
                                    2),
                              NA)

    results_labels <- as.data.frame(apply(results_labels,
                                          2,
                                          as.character),
                                    stringsAsFactors = FALSE)
    results_labels <- rbind(results_labels,
                            c("pr_adj", "Adjusted prevalence ratio"),
                            c("pvalue_adj", "Adjusted p value"),
                            c("paf_adj", "Adjusted Population Attributable Fraction (%)"))
  }


  # Organize the final results

  results$comp_index <- comp_index
  results_labels <- apply(results_labels,
                          2,
                          as.character)
  results_labels <- rbind(results_labels,
                          c("comp_index",
                            "Components index"))

  pcapaf_results <- list(results = results,
                         results_labels = results_labels)

  rm(leveldata, leveldata_model, wgt, packages, newpackages, nclasses, noutcomes,
     confcomplcases, outcomescomplcases, indexcomplcases, n_total_classes_temp,
     n_total_classes, n_classes_temp, n_classes, n_total_outcomes_temp, n_total_outcomes,
     n_outcomes_temp, n_outcomes, tab_wgt_classes, prop_classes_pop_temp, prop_classes_pop,
     comp_index, model, n_classes_outcomes, prop_outcomes_pop, prop_outcomes_classes,
     prop_classes_outcomes, pr, pvalue, paf, results, results_labels)
  if (adjustment == TRUE) {rm(model_adj, pr_adj, pvalue_adj)}
  gc()


  # End the function and return the final results

  return(pcapaf_results)
}
```

The `fun06_pafregionfun` function computes the confirmatory statistical analysis of the study, for Portugal and by region (NUTS II). This function has as dependency the `fun05_pcapaffun` function.

It has as arguments:

Arguments	Default	Description
`fun`	`fun05_pcapaffun`	Function `fun05_pcapaffun` to be computed for each region (Portugal and NUTS II)
`x`	pcaclasses	Output object from function `fun04_pcaclassesfun`

Arguments	Default	Description
`level`	Sem pré-definição	Level of analysis to be performed (whether by components or by classes)
`weights`	`TRUE`	Logical argument (`TRUE` or `FALSE`), for the use (or not) of the weights in the analysis (respectively)
`adjustment`	`TRUE`	Logical argument (`TRUE` or `FALSE`), to adjust (or not) to other components in the analysis (respectively)

It has as outputs:

- *list* containing two *data frames*:
  - `results`: computed statistics,
  - `results_labels`: description of the statistics listed in `results`.

```
fun06_pafregionfun <- function(fun = fun05_pcapaffun,
                               x = pcaclasses,
                               level = c("components",
                                         "classes"),
                               weights = TRUE,
                               adjustment = TRUE) {


  #### From the results of the fun05_pcapaffun function, make the analysis for Portugal
  #### and the different regions


  # Prepare the analysis

  regions <- as.character(x$data$others$REGION)
  strata  <- c("Portugal",
               unique(regions))
  resultsregion <- list(results = NULL,
                        results_labels = NULL)


  # Analysis for Portugal and regions

  for (i in 1:length(strata)) {
    if (strata[i] == "Portugal") {
      results_temp <- fun(x = x,
                          level = level,
                          weights = weights,
                          adjustment = adjustment)
      results_temp$results <- cbind(REGION = rep("Portugal",
                                                 dim(results_temp$results)[1]),
                                    results_temp$results)
      gc()
    } else {
      indexregion <- regions == strata[i]
      .x <- x
      .x$data$classes <- NULL
      .x$data$classes_model <- NULL
      .x$data$components <- NULL
```

```
      .x$data$diseases <- NULL
      .x$data$outcomes <- NULL
      .x$data$conf <- NULL
      .x$data$others <- NULL
      .x$data$classes <- x$data$classes[indexregion, ]
      .x$data$classes_model <- x$data$classes[indexregion, ]
      .x$data$components <- x$data$components[indexregion, ]
      .x$data$diseases <- x$data$diseases[indexregion, ]
      .x$data$outcomes <- x$data$outcomes[indexregion, ]
      .x$data$conf <- x$data$conf[indexregion, ]
      .x$data$others <- x$data$others[indexregion, ]
      results_temp <- fun(x = .x,
                          level = level,
                          weights = weights,
                          adjustment = adjustment)
      results_temp$results <- cbind(REGION = rep(strata[i],
                                                 dim(results_temp$results)[1]),
                                    results_temp$results)
      rm(.x)
      gc()
    }
    resultsregion$results <- rbind(resultsregion$results, results_temp$results)
  }

  resultsregion$results_labels <- data.frame(Code = c("REGION",
                                                      as.character(
                                                        results_temp$results_labels[, 1]
                                                      )),
                                             Label = c("Region (Portugal or NUTS II)",
                                                       as.character(
                                                         results_temp$results_labels[, 2]
                                                       )))

  rm(regions, strata, results_temp)
  gc()


  # End the function and return the final results

  return(resultsregion)
}
```

The `fun07_paflevelsfun` function computes the confirmatory statistical analysis of the study, by components and by classes. This function has as dependencies the `fun05_pcapaffun` and `fun06_pafregionfun` functions.

It has as arguments:

Arguments	Default	Description
`fun1`	`fun06_pafregionfun`	Function `fun06_pafregionfun` to be computed for each level of analysis (components and classes)
`fun2`	`fun05_pcapaffun`	Function `fun05_pcapaffun` to be computed for each region (Portugal and NUTS II)

Arguments	Default	Description
`x`	pcaclasses	Output object from function `fun04_pcaclassesfun`
`weights`	`TRUE`	Logical argument (`TRUE` or `FALSE`), for the use (or not) of the weights in the analysis (respectively)
`adjustment`	`TRUE`	Logical argument (`TRUE` or `FALSE`), to adjust (or not) to other components in the analysis (respectively)
`workingdirectory`	getwd()	Name of the folder previously passed to the `datafile` argument of the function `fun01_readandrecodefun` (between quotation marks and with the \\ or / symbol separating each level, eg: "C:\INS2014")

It has as outputs:

- *list* containing two *lists*, `components` and `classes`, each containing two *data frames*:
  - `results`: computed statistics, for each level of analysis,
  - `results_labels`: description of the statistics listed in `results`;
- four `.csv` files with computed statistics, for analysis by components and classes, and their descriptions, saved in the folder passed to the `workingdirectory` argument.

```
fun07_paflevelsfun <- function(fun1 = fun06_pafregionfun,
                               fun2 = fun05_pcapaffun,
                               x = pcaclasses,
                               weights = TRUE,
                               adjustment = TRUE,
                               workingdirectory = "C:\\INS2014") {


  #### From the results of the function fun06_pafregionfun, make the analysis for the
  #### components and classes


  # Open the appropriate working directory

  setwd(workingdirectory)


  # Analysis for components

  results_components <- fun1(fun = fun2,
                             x = x,
                             level = "components",
                             weights = weights,
                             adjustment = adjustment)
  gc()


  # Analysis for classes
```

```
  results_classes <- fun1(fun = fun2,
                          x = x,
                          level = "classes",
                          weights = weights,
                          adjustment = adjustment)
  gc()


  # Organize the results

  results <- list(components = results_components,
                  classes = results_classes)


  # Save the results to .csv files

  write.csv(results_components$results,
            "results_components.csv",
            row.names = FALSE)
  write.csv(results_components$results_labels,
            "results_components_labels.csv",
            row.names = FALSE)
  write.csv(results_classes$results,
            "results_classes.csv",
            row.names = FALSE)
  write.csv(results_classes$results_labels,
            "results_classes_labels.csv",
            row.names = FALSE)

  results_comp_classes_labels <- data.frame(Code_components =
                                              results_components$results_labels[, 1],
                                            Code_classes =
                                              results_classes$results_labels[, 1],
                                            Label =
                                              results_components$results_labels[, 2])
  results_comp_classes_labels$Label <- as.character(results_comp_classes_labels$Label)
  results_comp_classes_labels$Label[4] <- "Total number of individuals (components/classes)"
  write.csv(results_comp_classes_labels,
            "results_components_classes_labels.csv",
            row.names = FALSE)

  rm(results_components, results_classes, results_comp_classes_labels)
  gc()


  # End the function and return the final results

  return(results)
}
```

To perform the confirmatory analysis and compute the three previous functions, the following code was used:

```
pafresults <- fun07_paflevelsfun(fun1 = fun06_pafregionfun,
                                 fun2 = fun05_pcapaffun,
                                 x = pcaclasses,
                                 weights = TRUE,
```

```
                                    adjustment = TRUE,
                                    workingdirectory = "C:\\INS2014")
```

Outputs

Shiny app

The outputs of the final results were built in a *Shiny app*, available at *https://morbilidade.github.io/en/morbidity/*, for consultation in graphical or tabular form of all levels of analysis. This function has as dependencies the `DT` and `png` packages.

```
server <- function(input, output) {


  # If necessary install the DT and png package and load them

  packages <- c("DT", "png")
  newpackages <- packages[!(packages %in% installed.packages()[, "Package"])]
  if(length(newpackages)) install.packages(newpackages)
  library(DT); library(png)


  # Read and prepare data

  components_results <- read.csv("results_components.csv", fileEncoding = "UTF-8")
  components_results_labels <- read.csv("results_components_labels.csv", fileEncoding = "UTF-8")
  classes_results <- read.csv("results_classes.csv", fileEncoding = "UTF-8")
  classes_results_labels <- read.csv("results_classes_labels.csv", fileEncoding = "UTF-8")
  colors <- c("blue4",
              "brown4",
              "darkgoldenrod4",
              "darkolivegreen",
              "darkorchid4",
              "dodgerblue3",
              "green4",
              "lightslategrey")


  # Prepare some data for the UI

  regions <- unique(as.character(classes_results$REGION))[c(1,
                                                            2,
                                                            4,
                                                            5,
                                                            6,
                                                            3,
                                                            7,
                                                            8)]
  firstregion <- regions[1]
  outcomes <- unique(as.character(classes_results$outcomes))[c(1,
                                                               2,
                                                               3,
                                                               4,
```

```
                                                                    5,
                                                                    6,
                                                                    7,
                                                                    8,
                                                                    9,
                                                                    10,
                                                                    11,
                                                                    12,
                                                                    13,
                                                                    14)]
firstoutcome <- outcomes[1]
maxslider <- length(unique(classes_results$classes)) - 1
output$sliderInput <- renderUI({
  sliderInput(inputId = "ntop",
              label = "Top (graph):",
              min = 0,
              max = as.numeric(maxslider),
              value = 5,
              width = "110%")
})
output$selectInput_reg <- renderUI({
  selectInput(inputId = "reg",
              label = "Region:",
              choices = as.list(regions),
              selected = as.list(firstregion))
})
output$selectInput_out <- renderUI({
  selectInput(inputId = "out",
              label = "Morbidity indicators:",
              choices = as.list(outcomes),
              selected = as.list(firstoutcome),
              width = "110%")
})

observe({
  Region <- as.character(input$reg)
  Outcome <- as.character(input$out)
  Top_temp <- as.numeric(input$ntop)
  Level <- as.character(input$lev)
  Adjust <- as.character(input$adjust)


  # Outputs

  output$plot <- renderPlot({
    if (Level == "1") {
      if (Adjust == "1") {
        indexna <- !is.na(components_results$pvalue_adj) & components_results$pr_adj > 1
        index0001 <- components_results$pvalue_adj[indexna] == "<0.001"
        index005 <- as.numeric(
          as.character(
            components_results$pvalue_adj[indexna][!index0001]
          )
        ) <= 0.05
        index0001[!index0001] <- index005
        indexna[indexna] <- index0001
```

```
      indexna005 <- indexna
      indexna005 <- indexna &
        !is.na(components_results$pr_adj) &
        !is.na(components_results$paf_adj)
    } else if (Adjust == "2") {
      indexna <- !is.na(components_results$pvalue) & components_results$pr > 1
      index0001 <- components_results$pvalue[indexna] == "<0.001"
      index005 <- as.numeric(
        as.character(
          components_results$pvalue[indexna][!index0001]
        )
      ) <= 0.05
      index0001[!index0001] <- index005
      indexna[indexna] <- index0001
      indexna005 <- indexna
      indexna005 <- indexna &
        !is.na(components_results$pr) &
        !is.na(components_results$paf)
    }
    x <- components_results[indexna005, ]
  } else if (Level == "2") {
    if (Adjust == "1") {
      indexna <- !is.na(classes_results$pvalue_adj) & classes_results$pr_adj > 1
      index0001 <- classes_results$pvalue_adj[indexna] == "<0.001"
      index005 <- as.numeric(
        as.character(
          classes_results$pvalue_adj[indexna][!index0001]
        )
      ) <= 0.05
      index0001[!index0001] <- index005
      indexna[indexna] <- index0001
      indexna005 <- indexna
      indexna005 <- indexna &
        !is.na(classes_results$pr_adj) &
        !is.na(classes_results$paf_adj)
    } else if (Adjust == "2") {
      indexna <- !is.na(classes_results$pvalue) & classes_results$pr > 1
      index0001 <- classes_results$pvalue[indexna] == "<0.001"
      index005 <- as.numeric(
        as.character(
          classes_results$pvalue[indexna][!index0001]
        )
      ) <= 0.05
      index0001[!index0001] <- index005
      indexna[indexna] <- index0001
      indexna005 <- indexna
      indexna005 <- indexna &
        !is.na(classes_results$pr) &
        !is.na(classes_results$paf)
    }
    x <- classes_results[indexna005, ]
  }
  par(mar=c(0, 0, 0, 0))
  indexregion <- x$REGION == Region
  indexoutcome <- x$outcomes == Outcome
  index <- indexregion & indexoutcome
```

```
    if (sum(index) < 2 | Top_temp == 0) {
      plot.new()
    } else {
      if (sum(index) == 2) {
        ill <- x[index, ][1, ]
        data <- x[index, ][-1, ]
        data <- cbind(1, data)
        colnames(data) <- c("n_original",
                            names(data))
        colnames(ill) <- c("n_original",
                           names(data))
        Top <- 1
      } else {
        ill <- x[index, ][1, ]
        data <- x[index, ][-1, ]
        data <- cbind(n_original = 1:dim(data)[1],
                      data)
        Top <- ifelse(Top_temp > dim(data)[1],
                      dim(data)[1],
                      Top_temp)
      }
      toppr <- 1:Top
      toppaf <- 1:Top
      if (Level == "1") {
        if (Adjust == "1") {
          orderpr <- cbind(n_pr = 1:nrow(data),
                           data[, c("n_original",
                                    "components",
                                    "pr_adj",
                                    "comp_index")][order(data$pr_adj,
                                                         decreasing = TRUE), ])
          orderpaf <- cbind(n_paf = 1:nrow(data),
                            data[,c("n_original",
                                    "components",
                                    "paf_adj",
                                    "comp_index")][order(data$paf_adj,
                                                         decreasing = TRUE), ])
        } else if (Adjust == "2") {
          orderpr <- cbind(n_pr = 1:nrow(data),
                           data[,c("n_original",
                                   "components",
                                   "pr",
                                   "comp_index")][order(data$pr,
                                                        decreasing = TRUE), ])
          orderpaf <- cbind(n_paf = 1:nrow(data),
                            data[,c("n_original",
                                    "components",
                                    "paf",
                                    "comp_index")][order(data$paf,
                                                         decreasing = TRUE), ])
        }
        bottompr <- match(orderpaf$components[1:Top],
                          orderpr$components)
        bottompaf <- match(orderpr$components[1:Top],
                           orderpaf$components)
      } else if (Level == "2") {
```

```
  if (Adjust == "1") {
    orderpr <- cbind(n_pr = 1:nrow(data),
                     data[, c("n_original",
                              "classes",
                              "pr_adj",
                              "comp_index")][order(data$pr_adj,
                                                   decreasing = TRUE), ])
    orderpaf <- cbind(n_paf = 1:nrow(data),
                      data[, c("n_original",
                               "classes",
                               "paf_adj",
                               "comp_index")][order(data$paf_adj,
                                                    decreasing = TRUE), ])
  } else if (Adjust == "2") {
    orderpr <- cbind(n_pr = 1:nrow(data),
                     data[, c("n_original",
                              "classes",
                              "pr",
                              "comp_index")][order(data$pr,
                                                   decreasing = TRUE), ])
    orderpaf <- cbind(n_paf = 1:nrow(data),
                      data[, c("n_original",
                               "classes",
                               "paf",
                               "comp_index")][order(data$paf,
                                                    decreasing = TRUE), ])
  }
  bottompr <- match(orderpaf$classes[1:Top],
                    orderpr$classes)
  bottompaf <- match(orderpr$classes[1:Top],
                     orderpaf$classes)
}
indexpr <- sort(unique(c(toppr,
                         bottompr)))
indexpaf <- sort(unique(c(toppaf,
                          bottompaf)))
plotpr <- cbind(points_pr = 1:length(indexpr),
                orderpr[indexpr, ])
plotpaf <- cbind(points_paf = 1:length(indexpaf),
                 orderpaf[indexpaf, ])
lines_y_pr <- plotpr$points_pr
lines_y_paf <- plotpaf$points_paf[match(plotpr$n_original,
                                        plotpaf$n_original)]
lines_y_pr <- length(lines_y_pr) + 1 - lines_y_pr
lines_y_paf <- length(lines_y_paf) + 1 - lines_y_paf
change_index <- lines_y_pr - lines_y_paf
change <- ifelse(change_index > 0,
                 2,
                 ifelse(change_index < 0,
                        1,
                        3))
lines_y_pr[lines_y_pr <= (length(lines_y_pr) - Top)] <-
  lines_y_pr[lines_y_pr <= (length(lines_y_pr) - Top)] - 1
lines_y_paf[lines_y_paf <= (length(lines_y_paf) - Top)] <-
  lines_y_paf[lines_y_paf <= (length(lines_y_paf) - Top)] - 1
lines_y <- cbind(lines_y_pr,
```

```
                    lines_y_paf,
                    change)
    plotpaf$points_paf <- plotpr$points_pr <- lines_y_pr
    pr_x <- rep(-1.5, nrow(plotpr))
    pr_y <- plotpr$points_pr
    paf_x <- rep(1.5, nrow(plotpaf))
    paf_y <- plotpaf$points_paf
    if (Level == "1") {
      if (Adjust == "1") {
        pr_labels <- paste(plotpr$n_pr,
                           ". ",
                           plotpr$components,
                           " (",
                           sprintf("%.2f",
                                   plotpr$pr_adj),
                           ")",
                           sep = "")
        paf_labels <- paste(plotpaf$n_paf,
                            ". ",
                            plotpaf$components,
                            " (",
                            sprintf("%.2f",
                                    plotpaf$paf_adj),
                            ")",
                            sep = "")
      } else if (Adjust == "2") {
        pr_labels <- paste(plotpr$n_pr,
                           ". ",
                           plotpr$components,
                           " (",
                           sprintf("%.2f",
                                   plotpr$pr),
                           ")",
                           sep = "")
        paf_labels <- paste(plotpaf$n_paf,
                            ". ",
                            plotpaf$components,
                            " (",
                            sprintf("%.2f",
                                    plotpaf$paf),
                            ")",
                            sep = "")
      }
    } else if (Level == "2") {
      if (Adjust == "1") {
        pr_labels <- paste(plotpr$n_pr,
                           ". ",
                           plotpr$classes,
                           " (",
                           sprintf("%.2f",
                                   plotpr$pr_adj),
                           ")",
                           sep = "")
        paf_labels <- paste(plotpaf$n_paf,
                            ". ",
                            plotpaf$classes,
```

```
                          " (",
                          sprintf("%.2f",
                                  plotpaf$paf_adj),
                          ")",
                          sep = "")
  } else if (Adjust == "2") {
    pr_labels <- paste(plotpr$n_pr,
                       ". ",
                       plotpr$classes,
                       " (",
                       sprintf("%.2f",
                               plotpr$pr),
                       ")",
                       sep = "")
    paf_labels <- paste(plotpaf$n_paf,
                        ". ",
                        plotpaf$classes,
                        " (",
                        sprintf("%.2f",
                                plotpaf$paf),
                        ")",
                        sep = "")
  }
}
indextop <- 1:Top
indexbottom <- (Top + 1:nrow(plotpr))
length_unique <- length(unique(classes_results$classes))
blank <- length_unique - nrow(plotpr)
col_pr <- as.character(sapply(plotpr$comp_index,
                              function(x) colors[as.numeric(x)]))
col_paf <- as.character(sapply(plotpaf$comp_index,
                               function(x) colors[as.numeric(x)]))
plot(x = rep(c(-1, 1), each = nrow(plotpr)),
     y = c((lines_y_pr + blank),
           (lines_y_paf + blank)),
     pch = 20,
     xlim = c(-10,
              10),
     ylim = c(0,
              (length_unique + 4)),
     xaxt = 'n',
     yaxt = 'n',
     ann = FALSE,
     frame.plot = FALSE)
background <- readPNG("logo_morbilidade_bw.png")
rasterImage(background, par()$usr[1], par()$usr[3], par()$usr[2], par()$usr[4])
for (i in 1:nrow(lines_y)) {
  lines(x = c(-1, 1),
        y = (lines_y[i, 1:2] + blank),
        lty = lines_y[i, 3],
        col = col_pr[i])
}
points(x = c(-1, 1),
       y = c(length_unique + 2,
             length_unique + 2),
       pch = 20,
```

```
        lwd = 2)
lines(x = c(-1, 1),
      y = c(length_unique + 2,
            length_unique + 2),
      lty = 3,
      lwd = 2)

text(pr_x[indextop],
     (pr_y[indextop] + blank),
     pr_labels[indextop],
     pos = 2,
     cex = 1.1,
     col = col_pr[indextop])
text(paf_x[indextop],
     (paf_y[indextop] + blank),
     paf_labels[indextop],
     pos = 4,
     cex = 1.1,
     col = col_paf[indextop])
text(pr_x[indexbottom],
     pr_y[indexbottom] + blank,
     pr_labels[indexbottom],
     pos = 2,
     cex = 0.9,
     col = col_pr[indexbottom])
text(paf_x[indexbottom],
     paf_y[indexbottom] + blank,
     paf_labels[indexbottom],
     pos = 4,
     cex = 0.9,
     col = col_paf[indexbottom])
text(-1.5,
     (length_unique + 2),
     paste(ill[, 3],
           " (",
           sprintf("%.2f",
                   ill$pr),
           ")", sep = ""),
     pos = 2,
     cex = 1.2)
text(1.5,
     (length_unique + 2),
     paste(ill[, 3],
           " (",
           sprintf("%.2f",
                   ill$paf),
           ")", sep = ""),
     pos = 4,
     cex = 1.2)
text(-1.5,
     (length_unique + 4),
     "Prevalence ratios, p<0.05",
     pos = 2,
     cex = 1.5)
text(1.5,
     (length_unique + 4),
```

```
            "Population attributable fraction (%)",
            pos = 4,
            cex = 1.5)
      }
    },
    height = 1000,
    width = 1000)

    output$table <- DT::renderDataTable({
      if (dim(classes_results)[2] == 16) {
        variables <- c(1, 2, 4, 11, 13, 12, 7:10, 3, 5, 6)
      } else if (dim(classes_results)[2] == 19) {
        variables <- c(1, 2, 4, 11, 13, 12, 14, 16, 15, 7:10, 3, 5, 6)
      }
      if (Level == "1") {
        DT::datatable(
          data = components_results[components_results$REGION == Region &
                                      components_results$outcomes == Outcome,
                                    -(1:2)][, variables],
          colnames = as.character(components_results_labels[-(1:2), 2][variables]),
          rownames = NULL,
          options = list(paging = FALSE,
                         scrollX = TRUE)
        )
      } else if (Level == "2") {
        DT::datatable(
          data = classes_results[classes_results$REGION == Region &
                                   classes_results$outcomes == Outcome, -(1:2)][, variables],
          colnames = as.character(classes_results_labels[-(1:2), 2][variables]),
          rownames = NULL,
          options = list(paging = FALSE,
                         scrollX = TRUE)
        )
      }
    })
    output$subtitulo <- renderText({
      if (Region == regions[1]) {
        paste(Outcome, "in", Region, sep = " ")
      } else {
        paste(Outcome, "in the", Region, "region", sep = " ")
      }
    })
    output$subtitulo_2 <- renderText({
      if (Level == "1") {
        paste("Level 1: Components (groups of diseases)")
      } else if (Level == "2") {
        paste("Level 2: Classes (combinations of diseases within groups)")
      }
    })
  })
}


ui <- fluidPage(
```

```
# Inputs

fluidRow(
  br()
),
fluidRow(
  column(2,
         img(src = "logo_morbilidade.png",
             height = 100),
         align = "center"
  ),
  column(10,
         fluidRow(
           column(2,
                  img(src = "logo_ispup.svg",
                      height = 40),
                  align = "right"
           ),
           column(9,
                  img(src = "logos_pt_sns_arsn_aces_usp.png",
                      height = 40),
                  align = "left"
           )
         ),
         fluidRow(
           br()
         ),
         fluidRow(
           column(2,
                  h6("With the collaboration of:"),
                  align = "right"
           ),
           column(2,
                  img(src = "logo_insa.png",
                      height = 40),
                  align = "center"
           ),
           column(2,
                  img(src = "logo_ine.png",
                      height = 40),
                  align = "left"
           )
         ))

),
fluidRow(
  br()
),
fluidRow(
  br()
),
fluidRow(
  column(3,
         selectInput(inputId = "lev",
                     label = "Level:",
                     choices = c("Level 1: Components" = "1",
```

```
                                "Level 2: Classes" = "2"),
                    selected = "1")
  ),
  column(4,
         uiOutput("selectInput_reg")
  ),
  column(1,
         br()
  ),
  column(4,
         selectInput(inputId = "adjust",
                     label = "Results adjusted for (graph):",
                     choices = c("Confounders and components" = "1",
                                 "Only confounders" = "2"),
                     selected = "1",
                     width = "110%")
  )
),
fluidRow(
  column(7,
         uiOutput("selectInput_out")
  ),
  column(1,
         br()
  ),
  column(4,
         uiOutput("sliderInput")
  )
),


# Outcome and region

fluidRow(
  column(12,
         h3(textOutput("subtitulo")),
         align = "center"
  )
),
fluidRow(
  column(12,
         h4(textOutput("subtitulo_2")),
         align = "center"
  )
),


# Outputs

fluidRow(
  tabsetPanel(type = "tabs",
              tabPanel("Graph",
                       plotOutput("plot")),
              tabPanel("Table",
                       DT::dataTableOutput("table"))
  ),
```

```
    align = "center"
  )
)


shinyApp(ui = ui,
         server = server)
```

Other outputs

The remaining outputs presented in the article (figures and tables) were computed with the following code (with the `questionr` and`formattable` packages as dependencies):

```
# If necessary install the questionr and formattable package and load them

packages <- c("questionr", "formattable", "htmltools", "webshot")
newpackages <- packages[!(packages %in% installed.packages()[, "Package"])]
if(length(newpackages)) install.packages(newpackages)
library(questionr); library(formattable); library("htmltools"); library("webshot")


# Table with diseases

columns <- c("Total", "n", "Prevalence (%)", "Weighted prevalence (%)")
data_d <- cbind(ins2014$data$others$ILL,
                ins2014$data$diseases[, -1])
label_d <- c(ins2014$labels$others_labels$Label[5],
             ins2014$labels$diseases_labels[-1, 2])
total_d <- sapply(data_d,
                  function(x) sum(!is.na(x)))
table_d <- sapply(data_d,
                  table)
n_d <- table_d[2, ]
prop_d <- round(prop.table(table_d, margin = 2)[2, ] * 100, digits = 2)
wtdtable_d <- apply(data_d, 2,
                    wtd.table,
                    weights = ins2014$data$others$WGT)
wtdprop_d <- round(prop.table(wtdtable_d,
                              margin = 2)[2, ] * 100,
                   digits = 2)
tab_diseases <- cbind(total_d,
                      n_d,
                      prop_d,
                      wtdprop_d)
row.names(tab_diseases) <- label_d
tab_diseases <- tab_diseases[order(-wtdprop_d), ]
colnames(tab_diseases) <- columns
Table1 <- formattable(as.data.frame(tab_diseases),
                      list(
                        "Weighted prevalence (%)" = color_bar("lavender")
                      ),
                      align = "l")
```

```r
# Table with outcomes

columns <- c("Total", "n", "Prevalence (%)", "Weighted prevalence (%)")
data_o <- ins2014$data$outcomes[, -1]
label_o <- ins2014$labels$outcomes_labels[-1, 2]
total_o <- sapply(data_o,
                  function(x) sum(!is.na(x)))
table_o <- sapply(data_o,
                  table)
n_o <- table_o[2, ]
prop_o <- round(prop.table(table_o,
                           margin = 2)[2, ] * 100,
                digits = 2)
wtdtable_o <- apply(data_o,
                    2,
                    wtd.table,
                    weights = ins2014$data$others$WGT)
wtdprop_o <- round(prop.table(wtdtable_o,
                              margin = 2)[2, ] * 100,
                   digits = 2)
tab_outcomes <- cbind(total_o,
                      n_o,
                      prop_o,
                      wtdprop_o)
row.names(tab_outcomes) <- label_o
tab_outcomes <- tab_outcomes[order(-wtdprop_o), ]
colnames(tab_outcomes) <- columns
Table2 <- formattable(as.data.frame(tab_outcomes),
                      list(
                        "Weighted prevalence (%)" = color_bar("lavender")
                      ),
                      align = "l")


# Table with PCA results

sum_pca <- round(pcamodel[[8]]$summary,
                 3)
colnames(sum_pca) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8")
rownames(sum_pca) <- c("Factor loadings",
                       "Proportion of variance",
                       "Cumulative proportion of variance",
                       "Proportion of variance explained",
                       "Cumulative proportion of variance explained")
Table3 <- formattable(as.data.frame(sum_pca))


# Table with PCA factor loadings

load_pca <- round(pcamodel[[8]]$loadings[1:17, ], 3)
row.names(load_pca) <- ins2014$labels$diseases_labels[-1, 2]
colnames(load_pca) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8")
load_pca <- load_pca[c(7, 8, 10, 14, 1, 2, 15, 16, 5, 9, 17, 12, 13, 3, 4, 6, 11), ]
Table4 <- formattable(as.data.frame(load_pca),
                      list(
                        PC1 = formatter("span",
```

```
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC2 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC3 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC4 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC5 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC6 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC7 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray"))),
                         PC8 = formatter("span",
                                        style = x ~ ifelse(x > 0.4 | x < -0.4,
                                                           style(color = "black",
                                                                 font.weight = "bold"),
                                                           style(color = "gray")))
                       ))


# Save tables as .png files

webshot::install_phantomjs()
export_formattable <- function(f,
                               file, width = 750,
                               height = NULL,
                               background = "white",
                               delay = 0.2) {
  w <- as.htmlwidget(f,
                     width = width,
                     height = height)
  path <- html_print(w,
                     background = background,
                     viewer = NULL)
```

```
  url <- paste0("file:///",
                gsub("\\\\", "/",
                     normalizePath(path)))
  webshot(url,
          file = file,
          selector = ".formattable_widget",
          delay = delay)
                                }

export_formattable(f = Table1,
                   file = "Table1.png")

export_formattable(f = Table2,
                   file = "Table2.png")

export_formattable(f = Table3,
                   file = "Table3.png")

export_formattable(f = Table4,
                   file = "Table4.png")
```

Monitoring morbidity associated with chronic conditions:

Study of the coexistence of chronic diseases and their impact on specific morbidity indicators in the Portuguese National Health Survey 2014

Ivo Cruz

SEDE ADMINISTRATIVA

FACULDADE DE MEDICINA

INSTITUTO DE CIÊNCIAS BIOMÉDICAS ABEL SALAZAR